



LoriotPro V4.00 « Extended Edition » LUA Scripting Extension V1.0

Par Ludovic Lecointe



MCours.com

Copyright © 2005-2006 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

LoriotPro V4.00 « Extended Edition » LUA Scripting Extension V1.01	
Pré requis	6
Introduction	7
Besoin de traitement de l'information	7
Phase 1 : recherche documentaire	9
Phase 2 : trouver l'index du disque C :	12
Phase 3 : Collecter les valeurs	14
Phase 4 : faire le calcul	15
Phase 5 : afficher le résultat	17
Conclusion.....	18
Le Langage LUA	19
Intégration de LUA dans LoriotPro	21
SNMP Objet virtuel.....	23
Passage de paramètres.....	24
Objet snmp virtuel par défaut.....	35
Exception d'utilisation des objets SNMP virtuel	40
Utilisation des scripts LUA dans les ActiveView	41
Introduction	42
Type de scripts utilisable dans les ActiveView	43
Usage direct d'objets SNMP virtuels	44
Calculatrice scientifique.....	47
Lancement scheduler de script LUA	54
Script utilisé avec la fonction DoubleClick.....	57
Script associé au menu contextuel d'un objet graphique.....	58
Utilisation directe de script dans une ActiveView.....	61
Utilisation de Script d'initialisation pour les ActiveView Modal.....	65
Utilisation direct de script	66
A partir des menus du logiciel	67
A partir d'un Shortcut de host.....	74
ShortCut associant une ActiveView Modal et un Script LUA	76
A partir du Report Center.....	86
A partir du module de management des hosts (Bulk Configuration)	87
Plugin éditeur de script	93
Introduction	94
Zone d'édition du script	97
Zone d'édition des variables.....	99
Zone des messages systèmes	101
Zone d'affichage des graphiques (ActiveView)	102
Librairies LUA de LoriotPro	102
Introduction	103
Fichier de définition de variables système	104
LoriotPro (Ip) Librairie	109
Ip.Get	110
Ip.GetBRC	113
Ip.Set.....	114
Ip.SetBRC	116

Ip.GetNext	118
Ip.GetNextBRC	122
Ip.Gets	124
Ip.GetsBRC	126
Ip.GetRows	128
Ip.GetRowsBRC	131
Ip.GetPath	133
Ip.Trace	135
Ip.Print	136
Ip.Break	137
Ip.GetFirstIP	139
Ip.GetNextIP	141
Ip.GetIPInformation	142
Ip.SetIPOption	147
Ip.InsertNewContainer	150
Ip.InsertNewNetwork	151
Ip.InsertNewHost	157
Ip.GetUIDInformation	160
Ip.GetChildUID	165
Ip.atop	167
Ip.iptoa	168
Ip.HexToOID	169
Ip.GetIPFromMAC	170
Ip.FindName	171
Ip.FindIPFromDynamicDNS	172
Ip.SendEvent	174
Ip.SendExternEvent	176
Ip.IsLoadedMIBRef	178
Ip.IsRegisteredMIBRef	181
Ip.FindSNMPObject	182
Ip.GetFirstNetwork	183
Ip.GetNextNetwork	185
Ip.GetNetworkFromIP	187
Ip.GetNetworkInformation	189
Ip.GetFirstRouter	193
Ip.GetNextRouter	196
Ip.GetRouterInformation	197
Ip.LoadLibrary	199
Ip.SaveDirectoryAs	201
Ip.FreeLibrary	203
Ip.GetAllHostFromContainer	205
Ip.GetSNMPObjectDescription	207
Ip.GetTrapFilterList	208
Ip.GetTrapFilterInformation	212
Ip.GetTrapFilterActionList	216
Ip.GetTrapFilterActionInformation	219
Ip.GetEventList	222
Ip.GetEventInformation	225
Ip.GetEventActionList	227
Ip.GetEventFilterActionInformation	228
Ip.AckEvent	231
Ip.AckEventFilterAction	232
Ip.AckTrapFilter	233
Ip.AckTrapFilterAction	234
Ip.ClearIPTrapCounter	235
Ip.SetEventFilterActionDisable	236
Ip.SetTrapFilterActionDisable	237
Ip.SetEventFilterDisable	238
Ip.SetTrapFilterDisable	239
Ip.GetTrapFilterGlobalStatus	240

Ip.GetEventFilterGlobalStatus	241
Ip.SetTrapFilterGlobalStatus	242
Ip.SetEventFilterGlobalStatus.....	243
LoriotPro ActiveView (Ipav) Librairie	245
Ipav.Insert.....	246
Ipav.InsertObject	249
Ipav.GetInformation	252
Ipav.SetPosition	256
Ipav.SetSize	258
Ipav.GetPosition	259
Ipav.SetCoord	260
Ipav.RefreshMap	261
Ipav.SetClipart.....	262
Ipav.SetToBackGround	264
Ipav.SetToForGround.....	265
Ipav.SetDown	266
Ipav.SetUp.....	267
Ipav.SetName.....	268
Ipav.SetBrush.....	269
Ipav.SetFontName	271
Ipav.SetLineColor	273
Ipav.SetLineWidth	275
Ipav.SetOption.....	277
Ipav.SelectUID	284
Ipav.SetRef.....	285
Ipav.FindRef	287
Ipav.GetRef	288
Ipav.Delete	290
Ipav.DeleteAll	292
Ipav.SetSelected	293
Ipav.ClearSelected	295
Ipav.ClearAllSelected	297
Ipav.DeleteAllSelected	299
Ipav.SelectAllRef	301
Ipav.UnSelectAllRef.....	303
Ipav.FindFromAssociatedUID.....	304
Ipav.SelectFirst.....	305
Ipav.SelectNext	308
Ipav.GetGlobalOption	309
Ipav.SetGlobalOption	311
Ipav.Load	314
Ipav.Append	315
Ipav.Save	316
Ipav.ShiftPositionAllSelected	317
Ipav.ShiftPosition.....	319
Ipav.GetRectAllSelected.....	321
Ipav.SetOptionAllSelected.....	323
Ipav.SelectAll.....	325
Ipav.Copy	326
Ipav.CopyAllSelected	327
Ipav.Paste	328
Ipav.SetupCornerAllSelected	329
Ipav.SizeInRectAllSelected	331
Ipav.InsertAction.....	334
Ipav.InsertMenu.....	336
Ipav.DeleteAllMenu	340
Ipav.SetExpression.....	341
Ipav.DeleteAllFilter	344
Ipav.InsertFilter.....	345
LoriotPro (Ipw) Librairie	350

Ipw.MsgBox.....	351
Ipw.RequestString	352
Ipw.TimedMessageBox	353
Ipw.FileDialog.....	354
Ipw.MIBTablePicker.....	355
Ipw.GetColor	356
Ipw.GetIPFromDirectory	357
Ipw.GetIntervalTime	358
Ipw.EditString	359
Ipw.MIBExpertDlg.....	360
Ipw.SelectClipart	362
Ipw.GetUIDFromDirectory	363
Ipw.SetSnmptDlg.....	365
Ipw.MibTabDlg	366
Ipw.ShellExecute	367
Ipw.ReportGeneratorDlg.....	368
Ipw.LoadPlugin.....	369
Ipw.Console (beta)	370
Ipw.ConsoleAddText (beta)	371
Ipw.ConsoleKill (beta).....	372
Intégration du SLA dans LUA.....	374
LoriotPro (Ipsla) Librairie.....	388
Ipsla.GetLoriotProIDLList	389
Ipsla.GetSLAList.....	390
Ipsla.Compute	391
Exemple de code sous forme de fonction LUA.....	397
Gestion de périodes	398
Analyse de ressource SNMP.....	405
Librairies externe LUA.....	412
Exemple LUA_ZIP	413
Exemple LUA_ODBC	418
Exemple LUA_COM	420
Exemple LUA_SOCKET.....	424
Projet Lua_Wizard_LIB.....	426
Matrice de compatibilité des librairies LUA.....	429
ANNEXES	430
License	430
Lua 5.0 license.....	430
Lua 4.0 license.....	432

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-saxon, une connaissance de la langue anglaise est donc très souhaitable. La documentation de base du langage LUA est mise en annexe de cette documentation. 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 pouvez trouver des exemples et toute la documentation en ligne.

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

Un accès direct à la documentation du logiciel.

http://www.luteus.biz/Download/LoriotPro_Doc/V4/LoriotPro_Documentation.htm
http://www.loriotpro.com/Products/Online_DocumentationV3_ActiveView/ActiveView_Documentation.htm

Un accès aux principaux sites traitant de LUA.

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

...

Des portails SNMP

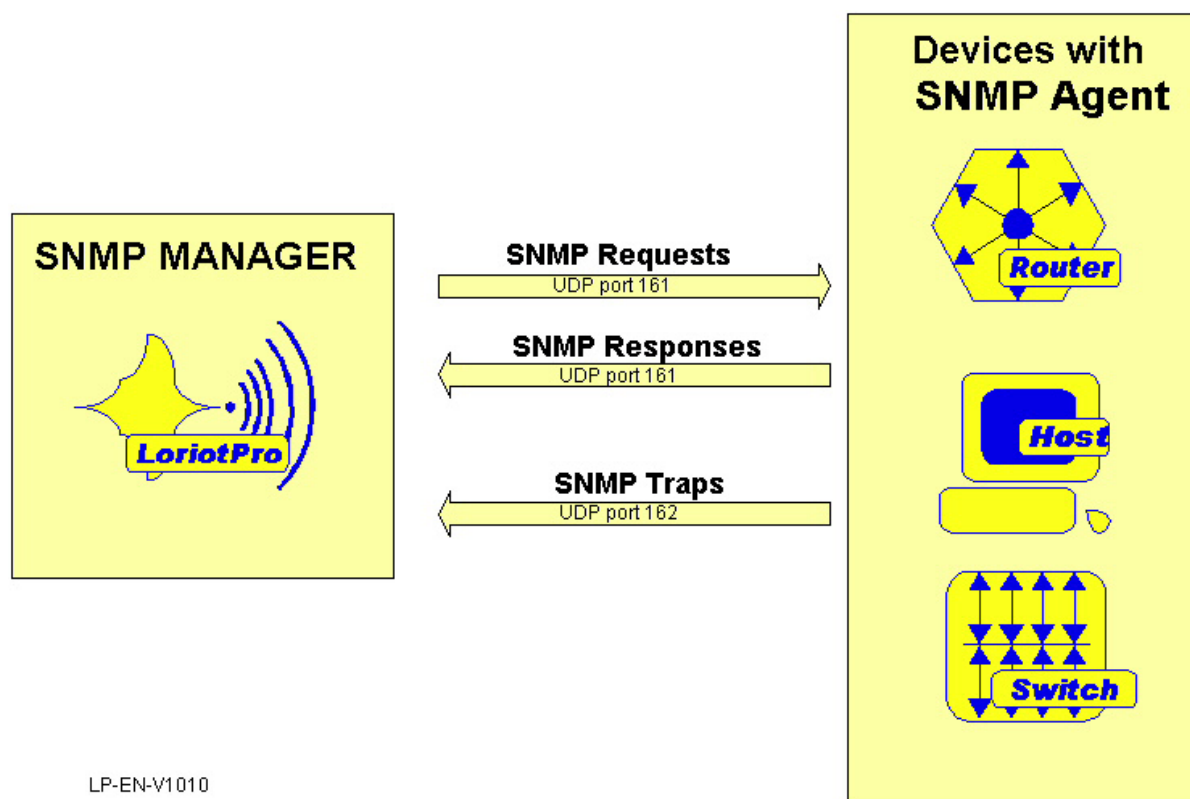
<http://www.snmplink.org/>
<http://www.snmpworld.com/>

Des portails partenaire de LUTEUS

http://www.loriotpro.com/Partners/PartnersList_EN.php

Introduction

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 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.



Besoin de traitement de l'information.

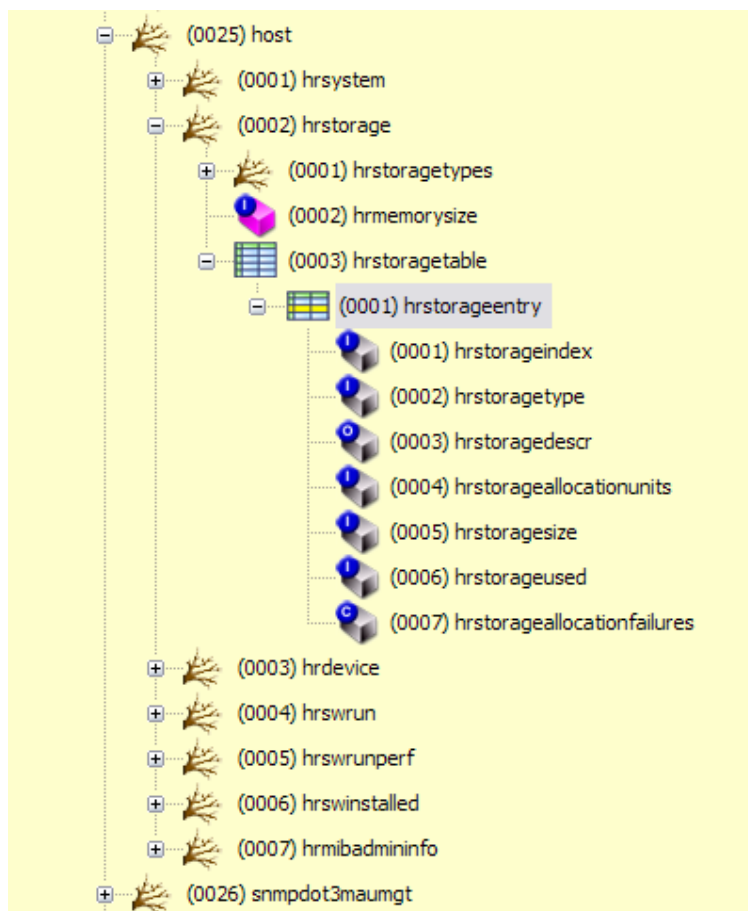
Le protocole SNMP n'intègre pas la notion de traitement mathématique ou logique des variables entre elles. Ce travail est laissé à la charge du logiciel d'administration du manager. Un exemple simple peut illustrer notre propos. Si nous désirons connaître le pourcentage d'utilisation du disque dur C : 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 standards 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.

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

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. 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

Objet SNMP	Description
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	"The type of storage represented by this entry."

OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.2.	
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 V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

The screenshot displays the LoriotPro V4.00 Extended Edition interface. The main window shows a tree view of MIB files under 'HOST-RESOURCES-MIB'. A report script window is open, displaying the output for 'rfc2790.mib'. The report includes the following information:

- Host Name: LoriotPro
- Host: 127.0.0.1
- Generated at: Fri Oct 28 11:54:56 2005
- MIB Definitions of **HOST-RESOURCES-MIB**
- Group: **hrsystem**
- Table 1: hrSystemDate
- Table 2: hrSystemInitialLoadDevice

Name	Value	Description
hrSystemDate	2005-10-28,11:54:56.0	The host's notion of the local date and time of day.
hrSystemInitialLoadDevice	0	The index of the hrDeviceEntry for the device from which this host is configured to load its initial operating system configuration (i.e., which operating system code and/or boot parameters). Note that writing to this object just changes the configuration that will be used the next time the operating system is loaded and does not actually cause

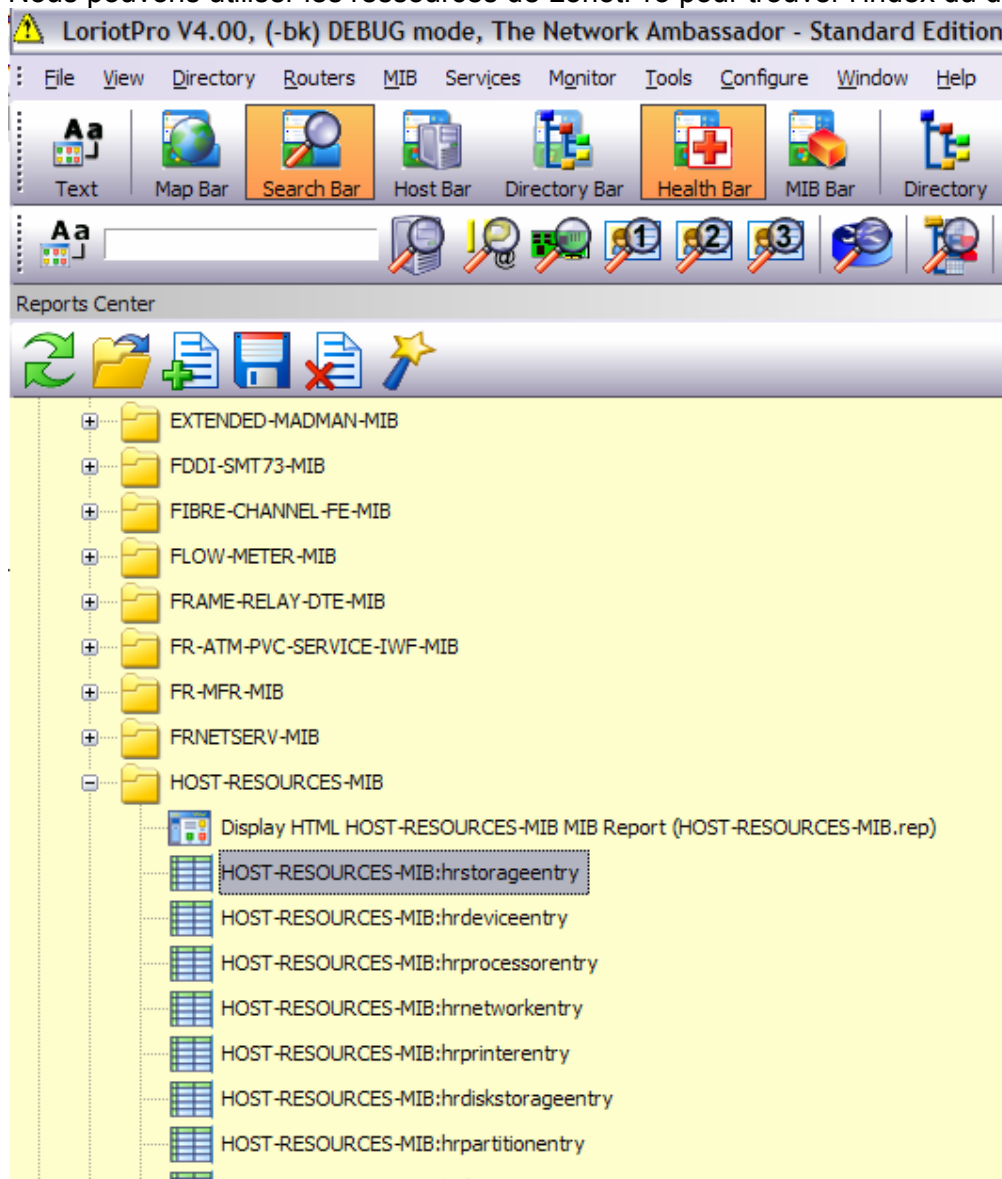
The interface also shows a taskbar with various system icons and a tasklist window at the bottom right.

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 pour cent.

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

Phase 2 : trouver l'index du disque C :

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



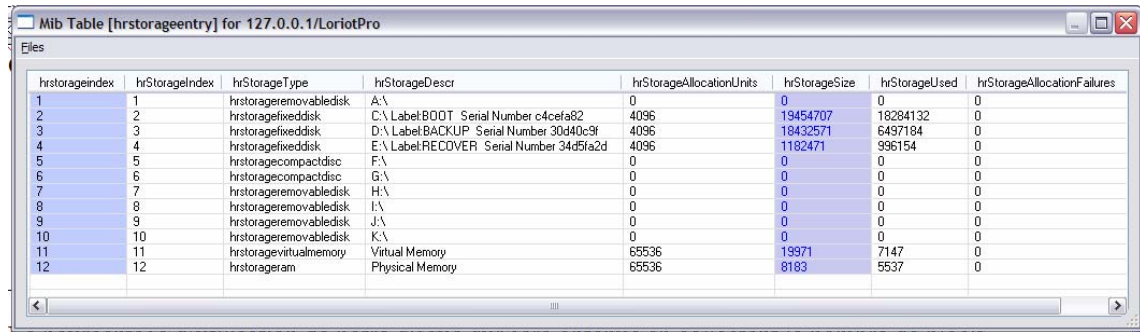
A partir du Report Center double cliquer sur le champ :

HOST-RESOURCES-MIB :hrstorageentry

LoriotPro contacte le host par défaut préalablement sélectionné dans la directory et affiche le contenu de la table.

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

LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA



Mib Table [hrstorageentry] for 127.0.0.1/LoriotPro

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

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

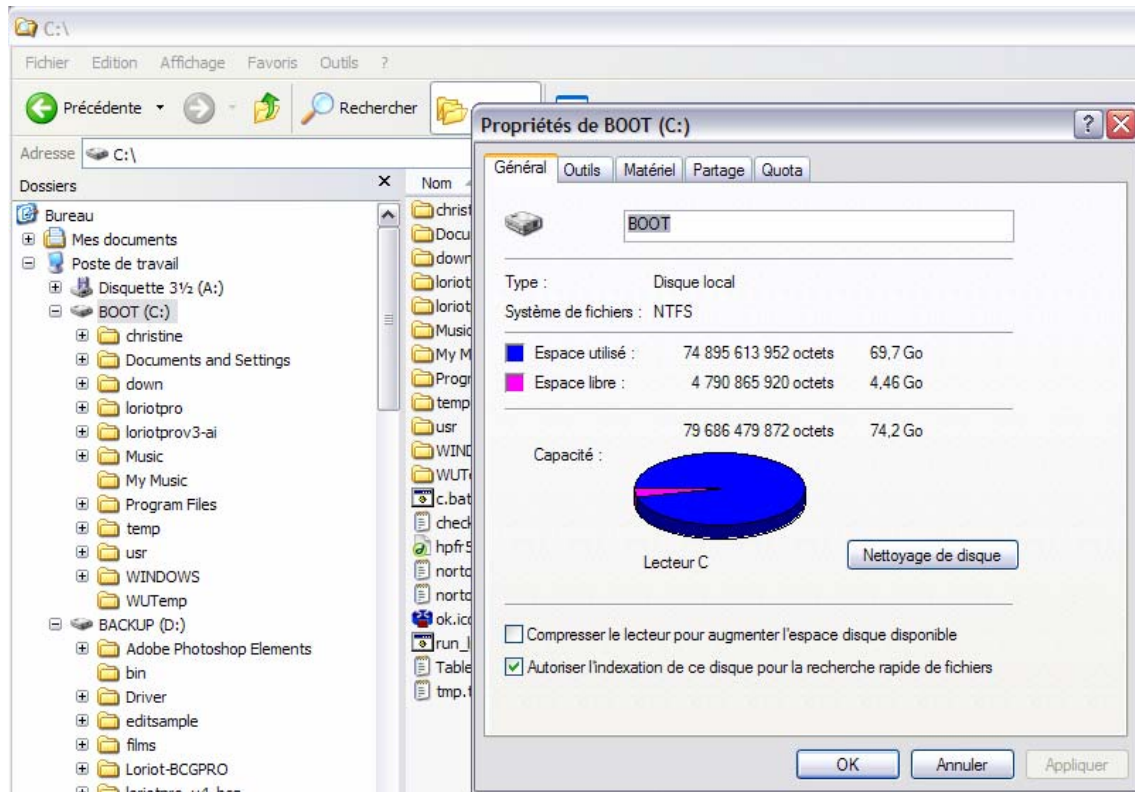
hrstoragesize.2 = 19454707

Phase 4 : faire le calcul

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

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 - hrstorageused.2) en respectant l'indexation du disque C : « .2 ».

Soit :

$$\text{hrstorageallocationunits.2} * (\text{hrstoragesize.2} - \text{hrstorageused.2}) =$$

$$4096 * (19454707 - 18284132) = 4794675200 \text{ octets}$$

Que nous pouvons transformer en Gigaoctet :

$$1024(\text{kilo}) * 1024(\text{mega}) * 1024(\text{giga}) = 1073741824$$

$$(4794675200) / 1073741824 = 4,4653 \text{ GigaOctets}$$

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 existant 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)

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

Conclusion

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

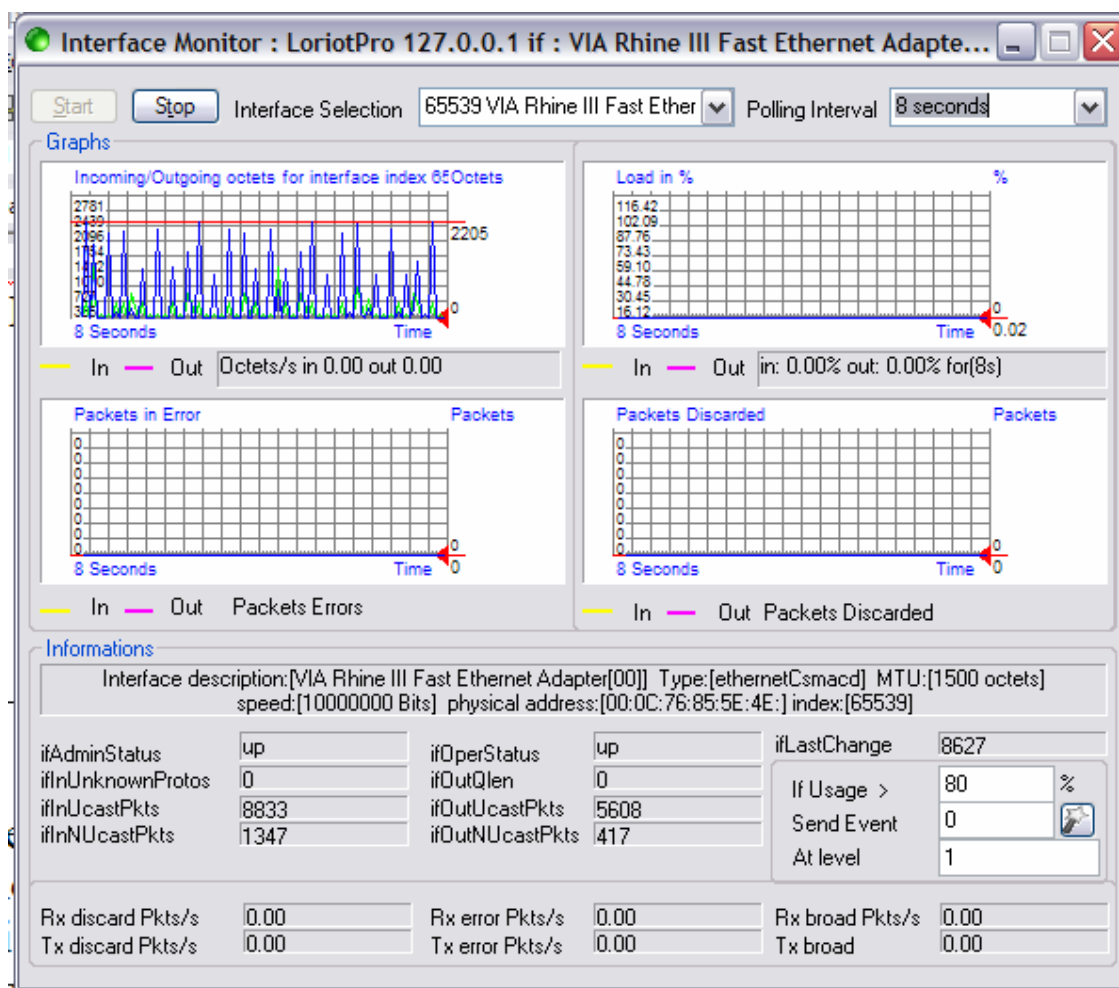
Un autre exemple simple est celui des séquences, en effet certaines variables SNMP retourne une valeur entière qui correspond en réalité à un statut. Il peut alors être nécessaire en fonction de la valeur retournée de collecter une autre variable. Dans ce cas une opération de tri ou de choix multiple 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 standards, 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 toutes sortes 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 script aux 5 phases nécessaires à l'intégration de collectes complexes sur les agents SNMP.

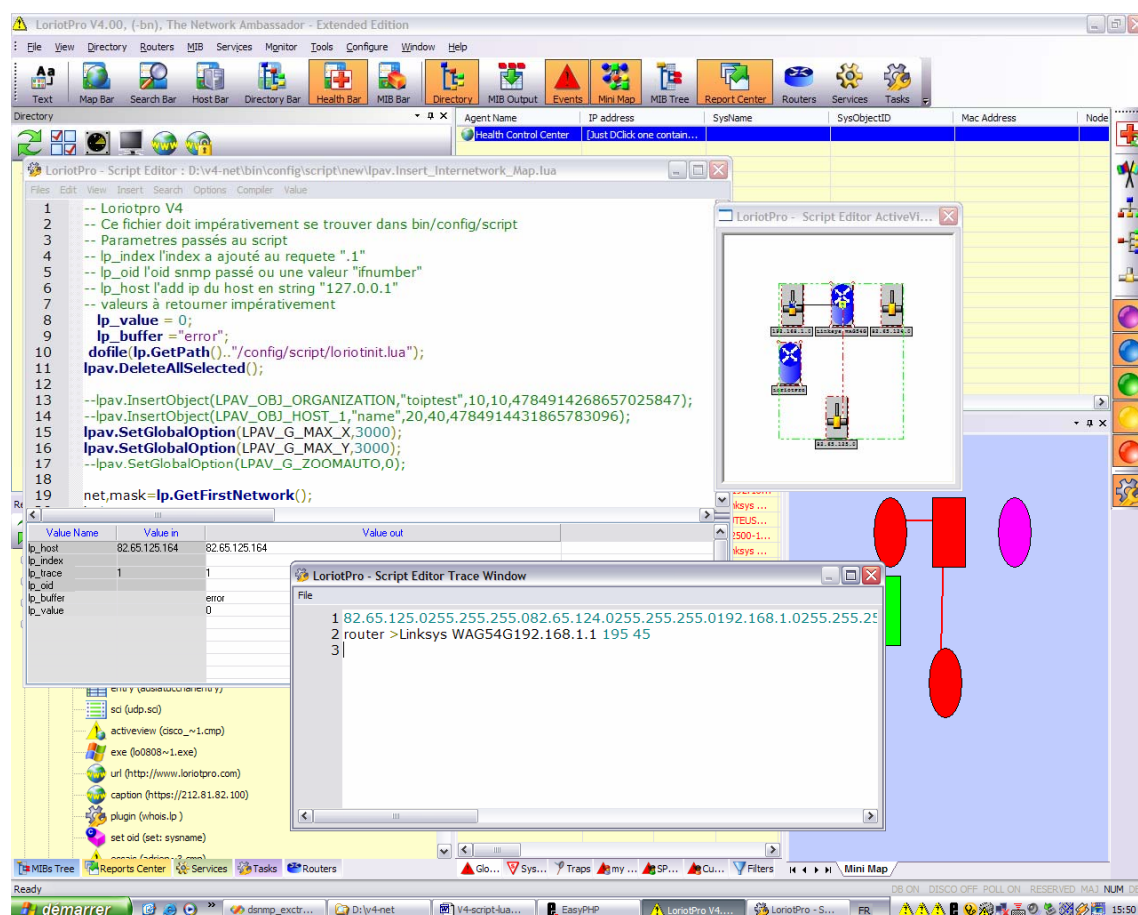
Le Langage LUA

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 intelligente.

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, situé dans le répertoire *c:\loriotpro-rep\bin\config\script*. Ces scripts sont utilisables sous la forme d'objet SNMP virtuel ou bien directement à partir d'une ActiveView, du « Report Center », de « ShortCut » de host, ou des différents menus du logiciel. Un environnement de développement est fourni pour permettre de créer, tester et lancer des scripts de tous types. En fonction du type d'utilisation des scripts il existe plusieurs bibliothèques de fonctions spécifiques. LoriotPro utilise la version 5.02 de LUA en mode dll (lua50.dll) ce qui permet l'utilisation de bibliothèques externes. La suite du document s'applique à montrer comment utiliser ces fonctions et bibliothèques.



L'environnement de développement intégré à LoriotPro en action.

SNMP Objet 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 » 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 » 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 fichier 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**.

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 suivant au script :

Paramètres	Descriptions
lp_host	L'adresse IP du host 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 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)

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_iod 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

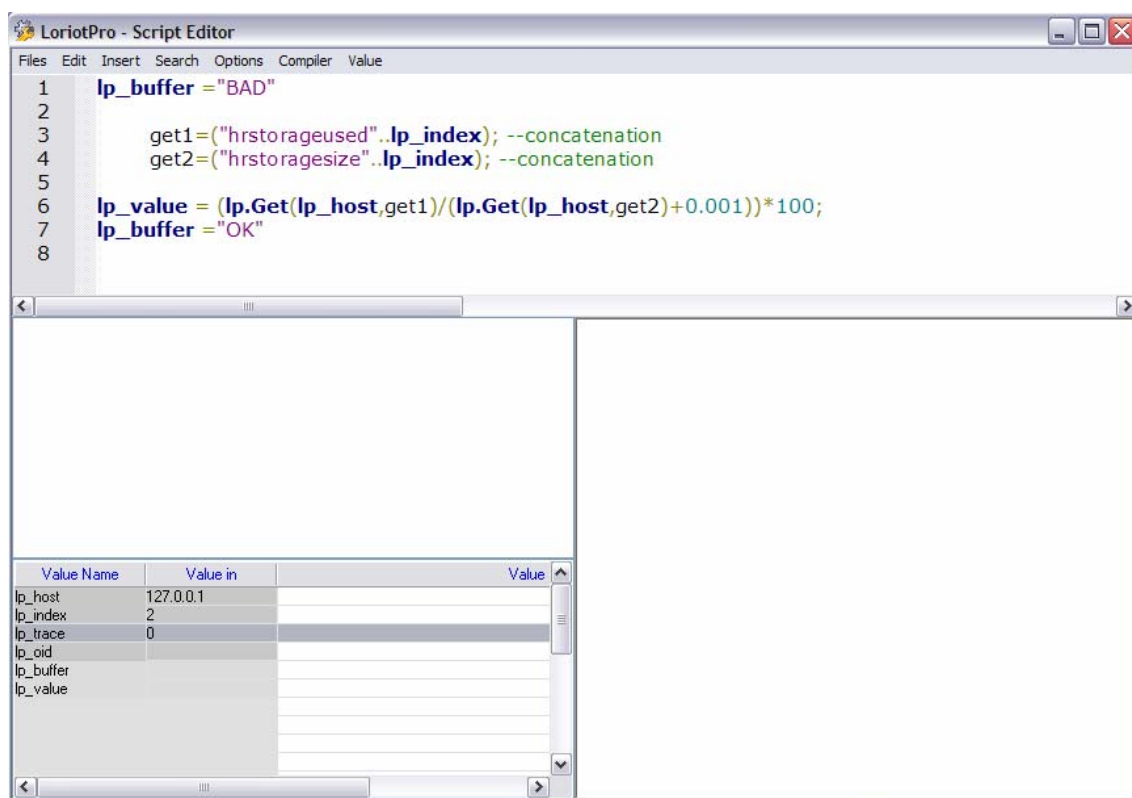
lp_value = (lp.Get(lp_host,get1)/(lp.Get(lp_host,get2)+0.001))*100;
lp_buffer ="OK"

<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 script editor contains the following Lua code:

```
1  lp_buffer = "BAD"
2
3      get1=("hrstorageused"..lp_index); --concatenation
4      get2=("hrstoragesize"..lp_index); --concatenation
5
6  lp_value = (lp.Get(lp_host,get1)/(lp.Get(lp_host,get2)+0.001))*100;
7  lp_buffer = "OK"
8
```

Below the script editor is a table with the following data:

Value Name	Value in	Value
lp_host	127.0.0.1	
lp_index	2	
lp_trace	0	
lp_oid		
lp_buffer		
lp_value		

La touche F5 permet de lancer le script

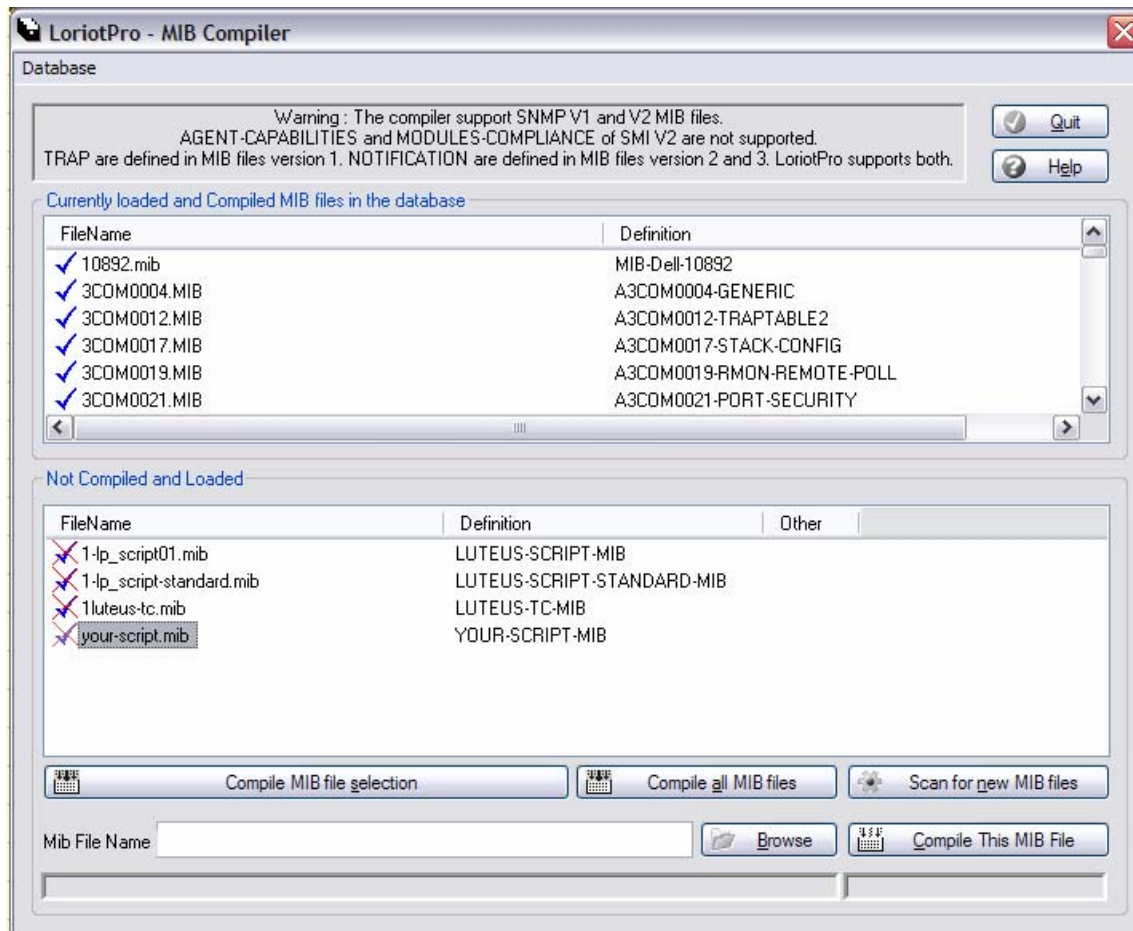
The screenshot shows the 'LoriotPro - Script Editor' window. The script content is as follows:

```
1  lp_buffer = "BAD"
2
3  get1=("hrstorageused"..lp_index); --concatenation
4  get2=("hrstoragesize"..lp_index); --concatenation
5
6  lp_value = (lp.Get(lp_host,get1) / (lp.Get(lp_host,get2)+0.001))*100;
7  lp_buffer = "OK"
8
```

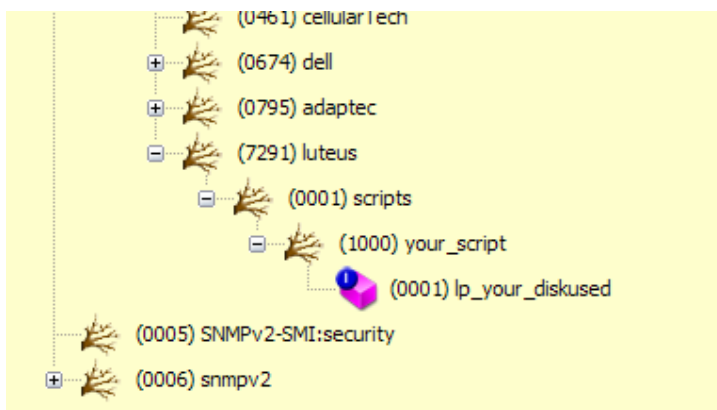
An 'Information' dialog box is displayed with the message 'Parsing Ended' and an 'OK' button.

Value Name	Value in	Value out
lp_host	127.0.0.1	127.0.0.1
lp_index	2	2
lp_trace	0	0
lp_oid		
lp_buffer		OK
lp_value		94.011104865573

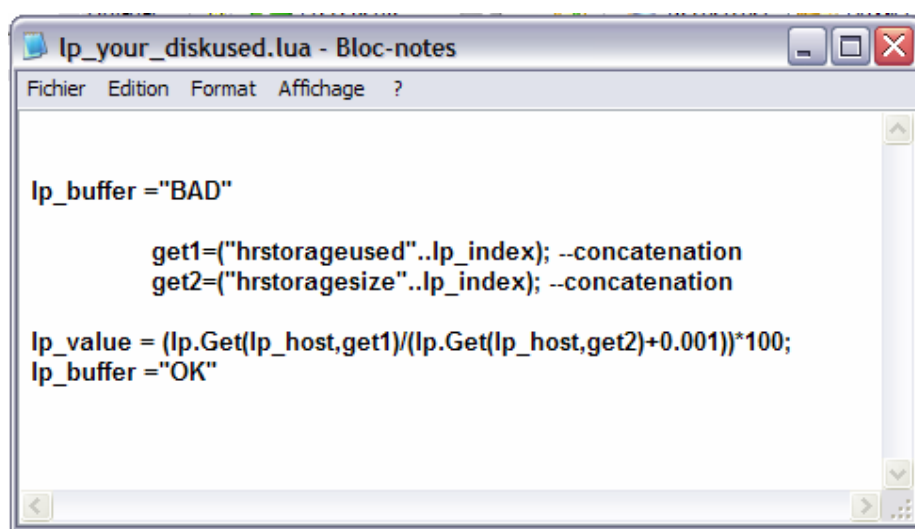
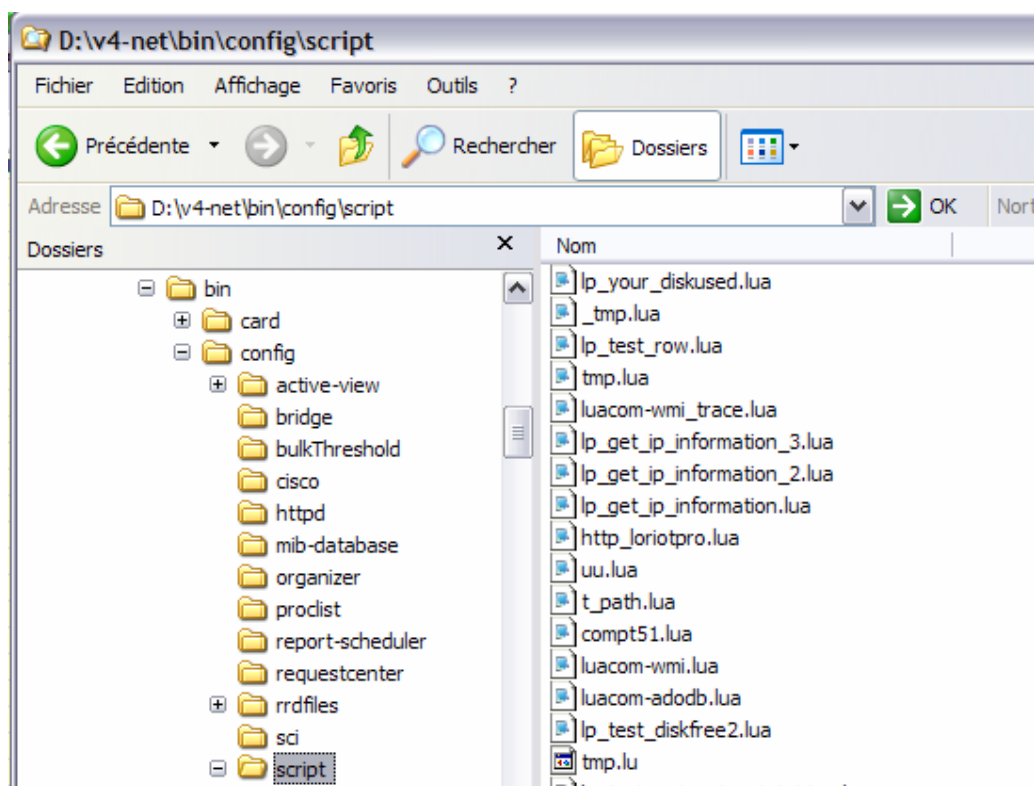
On place le fichier **your-script.mib** dans le répertoire *bin/mibs* et on lance le compilateur de MIB.



Après la compilation votre script est vu comme un objet SNMP.

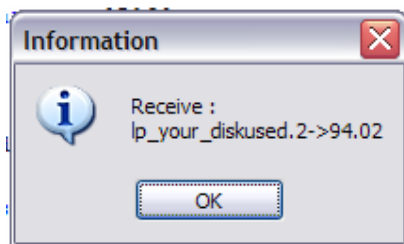
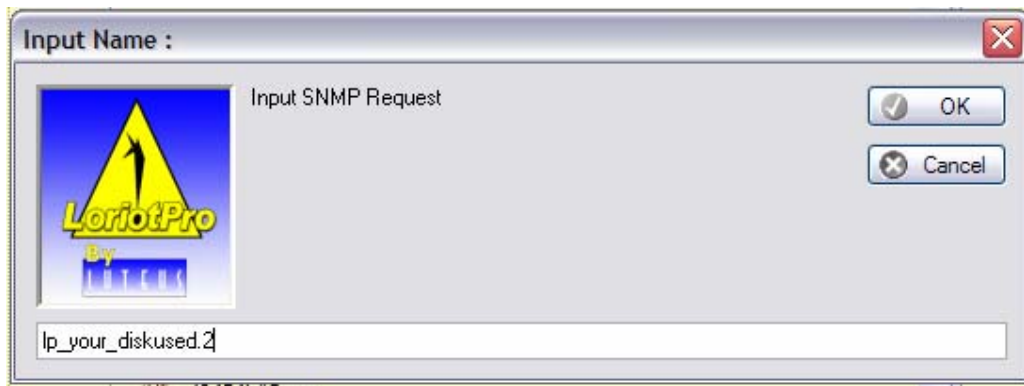


Le fichier « *lp_your_diskused.lua* » a été créé dans le répertoire *bin/config/script*



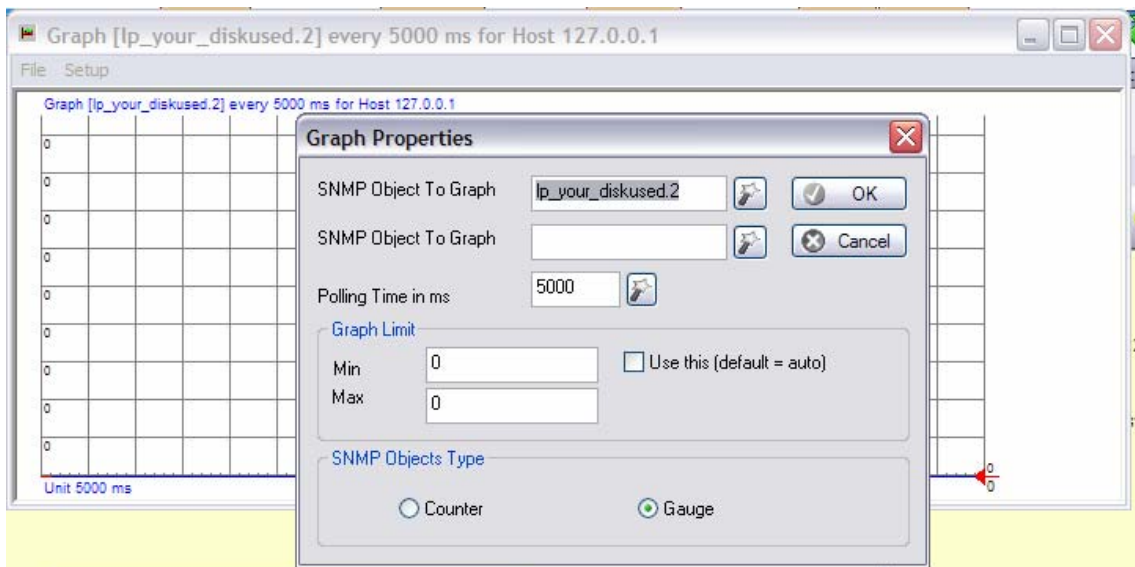
L'objet peut maintenant être utilisé comme un objet SNMP standard à travers tous les modules de LoriotPro.

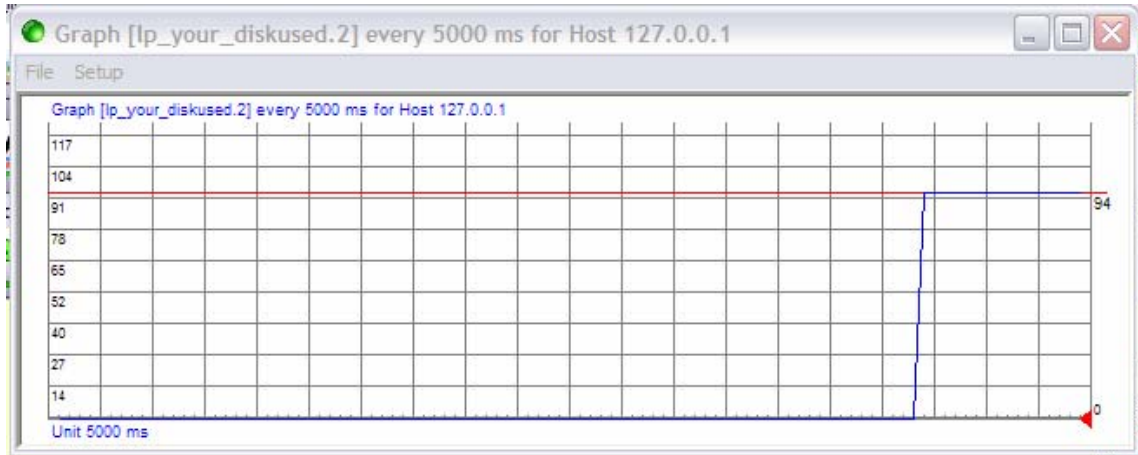
Attention cet objet est indexé et il faudra donc indiquer manuellement l'index du disque que nous désirons tester.



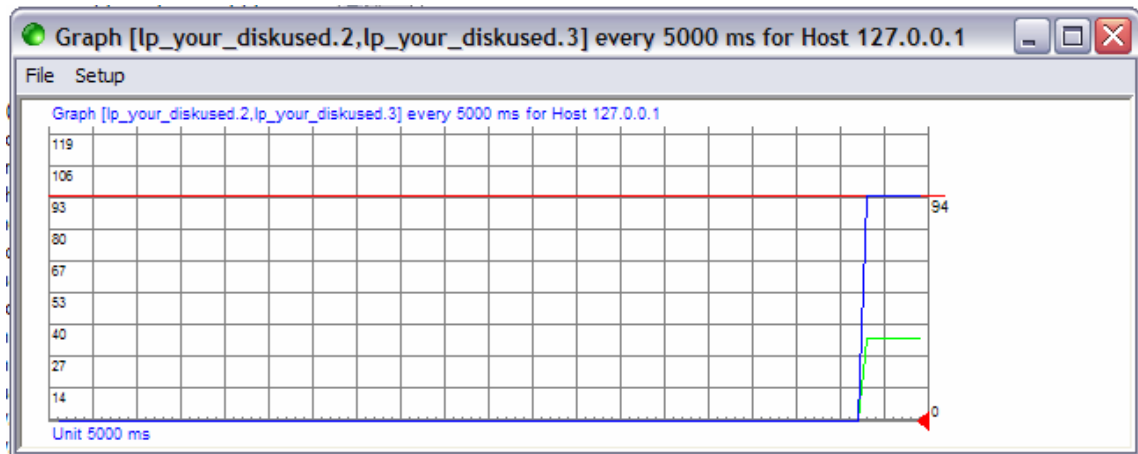
Dans ce module qui affiche le nom de l'objet SNMP reçu, nous avons *lp_your_diskused.2* qui n'est en réalité qu'un artefact par contre nous avons bien les 94,02 % d'utilisation de notre disque C :.

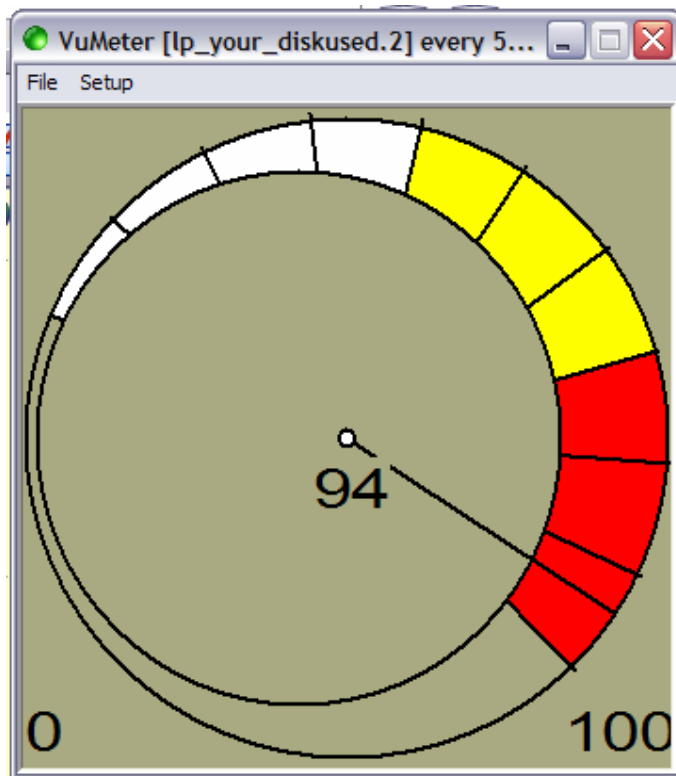
Nous pouvons l'utiliser dans un plugin de graph linéaire.





Des requêtes multiples peuvent être réalisées en changeant l'index.





Utilisation avec le plugin « VuMeter ».

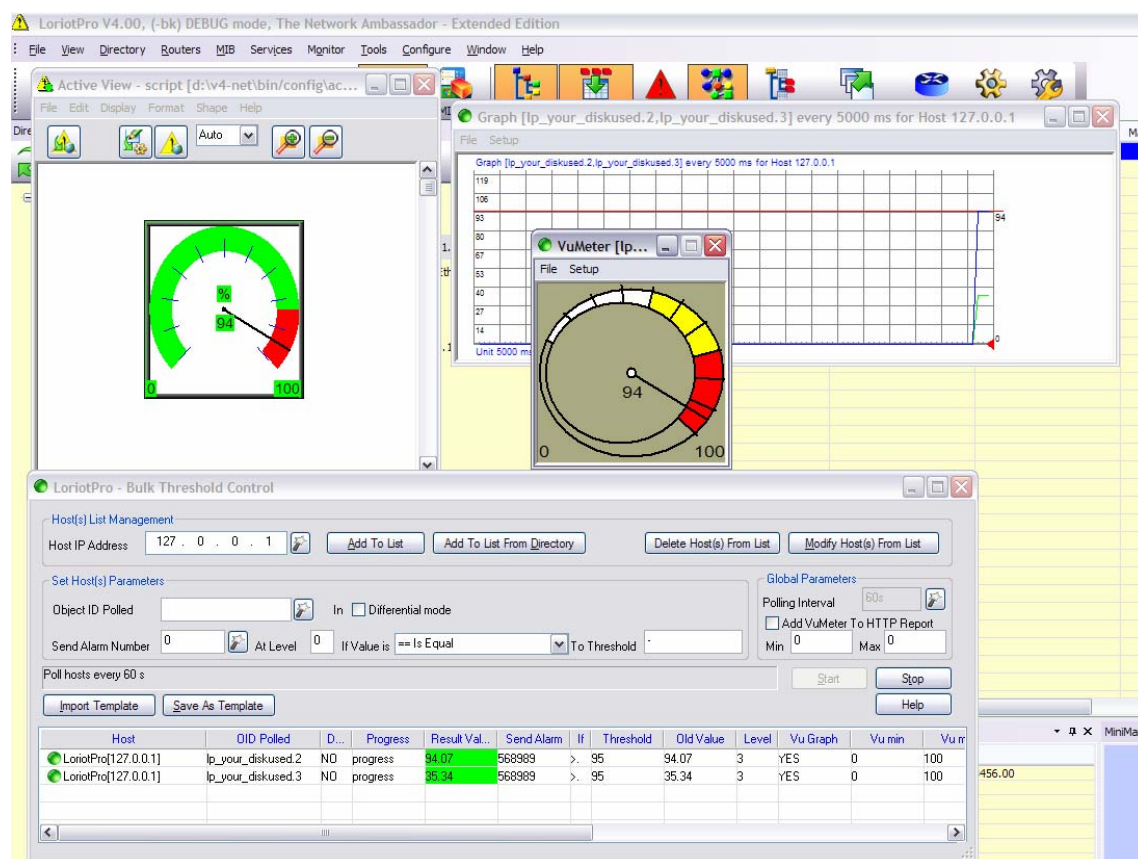
The image shows the 'LoriotPro - Bulk Threshold Control' window. It has several configuration sections:

- Host(s) List Management:** Host IP Address: 127.0.0.1. Buttons: Add To List, Add To List From Directory, Delete Host(s) From List, Modify Host(s) From List.
- Set Host(s) Parameters:** Object ID Polled: lp_your_diskused.3. In Differential mode. Send Alarm Number: 568989. At Level: 3. If Value is: >. Become: Greater than. To Threshold: 95.
- Global Parameters:** Polling Interval: 60s. Add VuMeter To HTTP Report. Min: 0. Max: 100.
- Buttons:** Start, Stop, Help.
- Table:** A table with columns: Host, OID Polled, D..., Progress, Result Val., Send Alarm, If, Threshold, Old Value, Level, Vu Graph, Vu min, Vu max, Last Polled, Alarm nb.

Host	OID Polled	D...	Progress	Result Val.	Send Alarm	If	Threshold	Old Value	Level	Vu Graph	Vu min	Vu max	Last Polled	Alarm nb
LoriotPro[127.0.0.1]	lp_your_diskused.3	NO	progress	95.30	568989	>	95		3	YES	0	100	Fri Oct 28 17:26:56 2005	
LoriotPro[127.0.0.1]	lp_your_diskused.2	NO	progress	94.04	568989	>	95	94.04	3	YES	0	100	Fri Oct 28 17:26:56 2005	

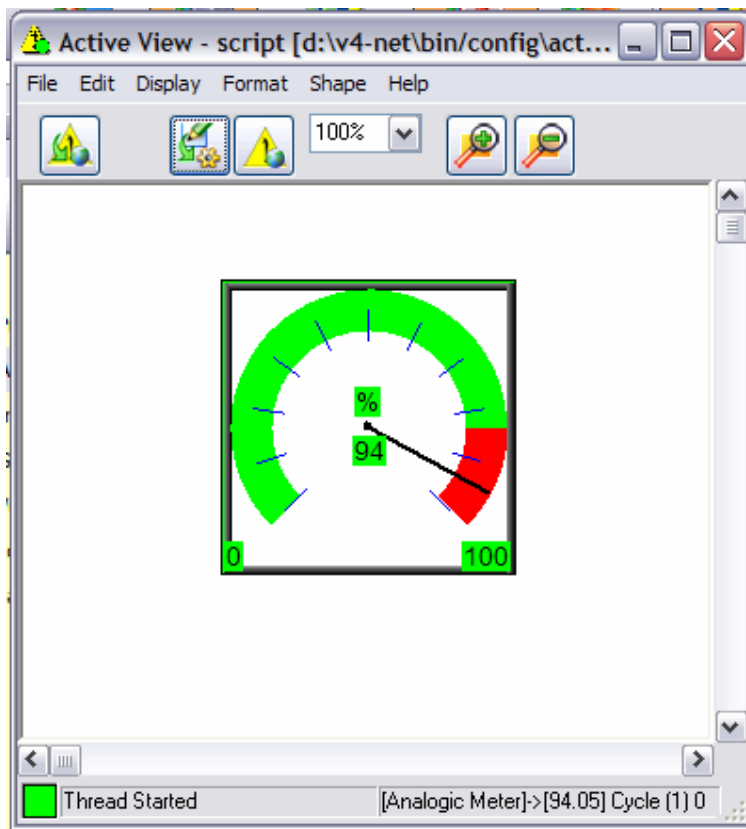
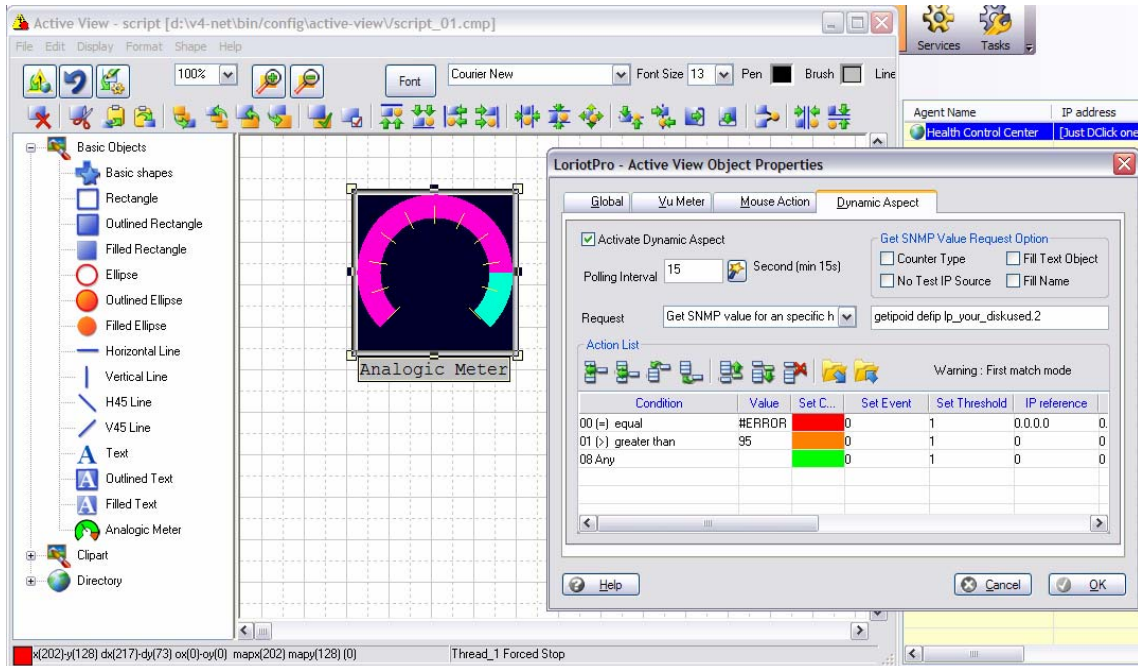
Utilisation avec le plugin « Bulk Threshold Control ».

LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA



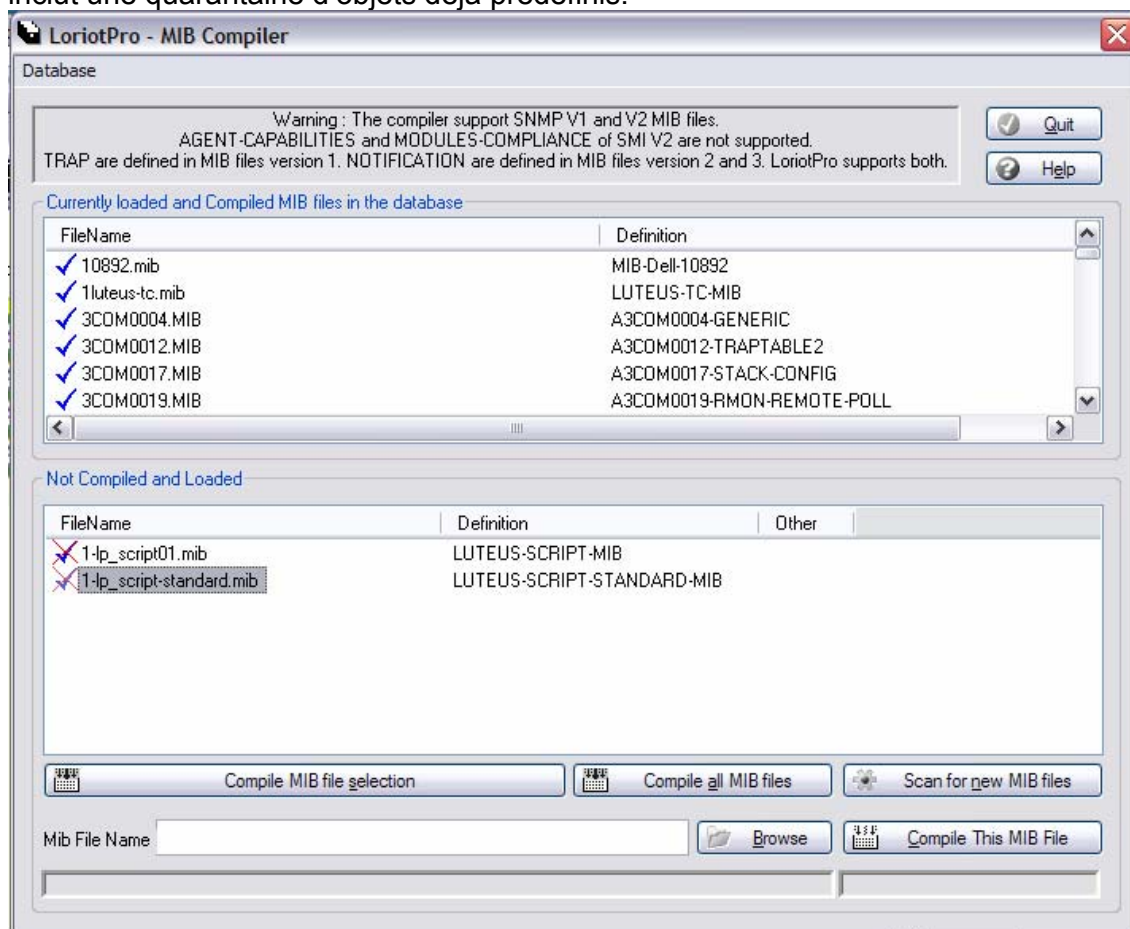
Le principe des objets snmp virtuels est simple, une requête avec l'OID du ou des objets virtuels est envoyée vers le host. Au retour du paquet (mode synchrone uniquement), le noyau de LoriotPro regarde les objets de type **lp_access_script** (logiquement à NUL) retournés par l'agent snmp et les remplace par le résultat des scripts. Les scripts peuvent eux aussi envoyer des requêtes vers l'agent snmp ou un ensemble d'agents snmp.

Les objets snmp virtuel sont bien sûr utilisables dans des ActiveView.

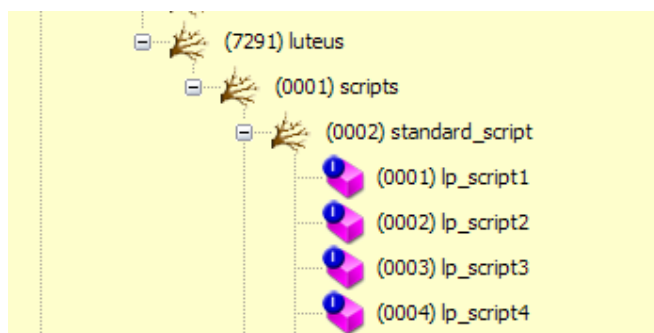


Objet snmp virtuel par défaut

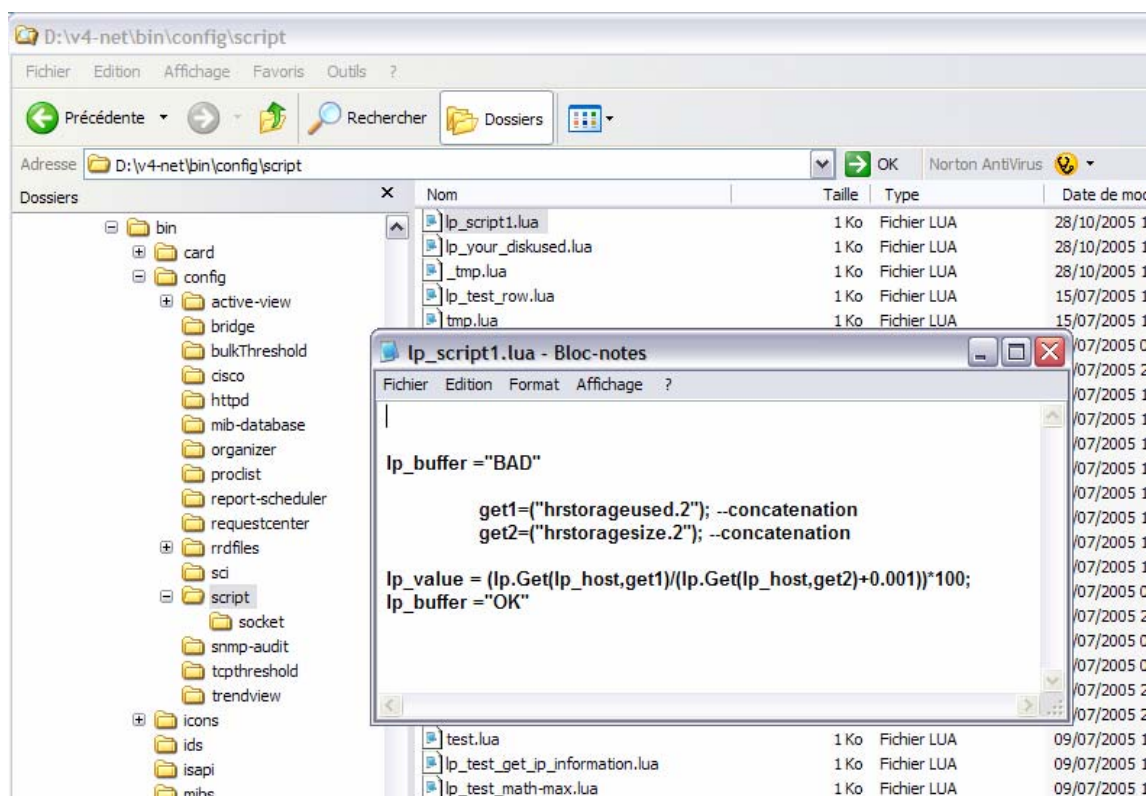
Pour simplifier la réalisation d'objet snmp virtuel utilisant des scripts LUA, le logiciel inclut une quarantaine d'objets déjà prédéfinis.



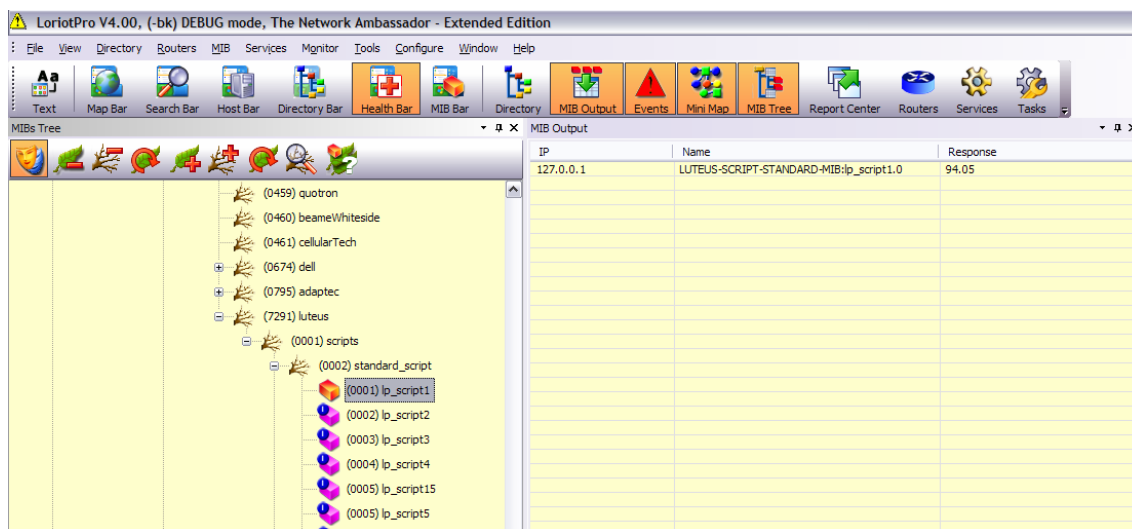
Il suffit de compiler la MIB LUTEUS-SCRIPT-STANDARD-MIB pour pouvoir disposer de 40 objets à définir simplement en créant les fichiers de script correspondant dans le répertoire bin/config/script.



Si vous voulez utiliser l'objet *lp_script1* créer simplement un script *lp_script1.lua* dans le répertoire bin/config/script et l'objet sera disponible.



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 caractère (« string ») voici la syntaxe à utiliser dans le fichier de MIB.

```
lp_your_string OBJECT-TYPE
    SYNTAX      OCTETSTRING
    ACCESS      lp_access_script
    STATUS      current
```

```
DESCRIPTION
    "return a string"
<LP_SCRIPT>

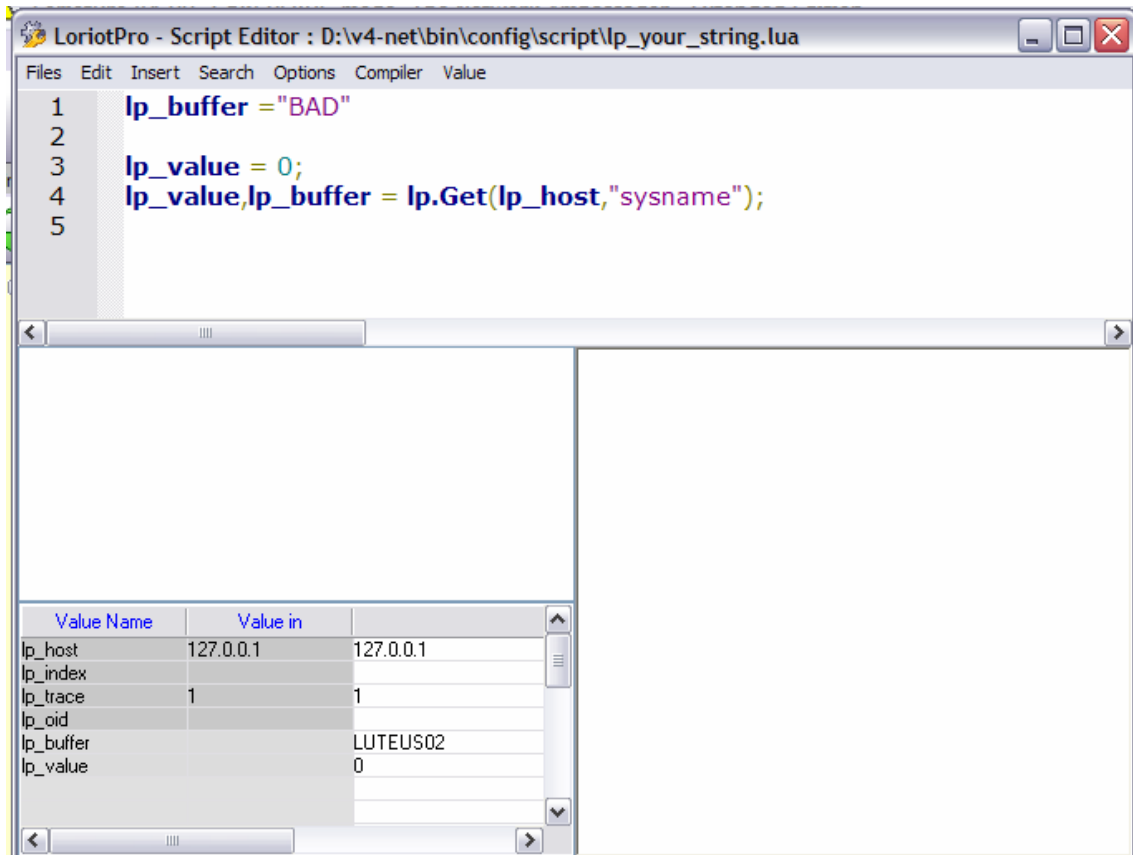
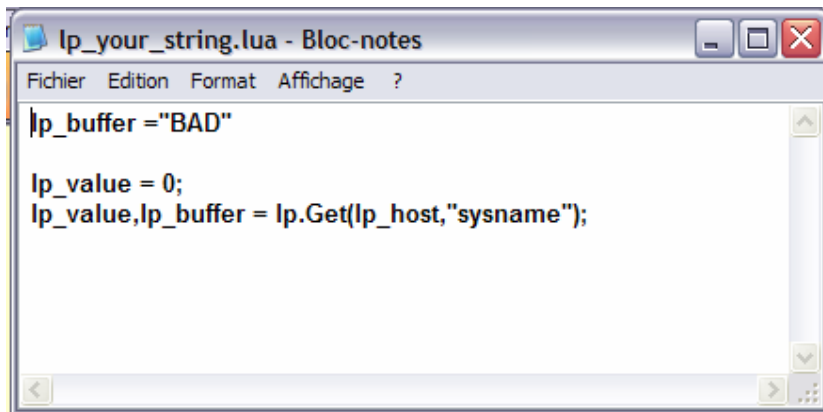
lp_buffer ="BAD"

lp_value, lp_buffer =lp.Get(lp_host,"sysname.0")

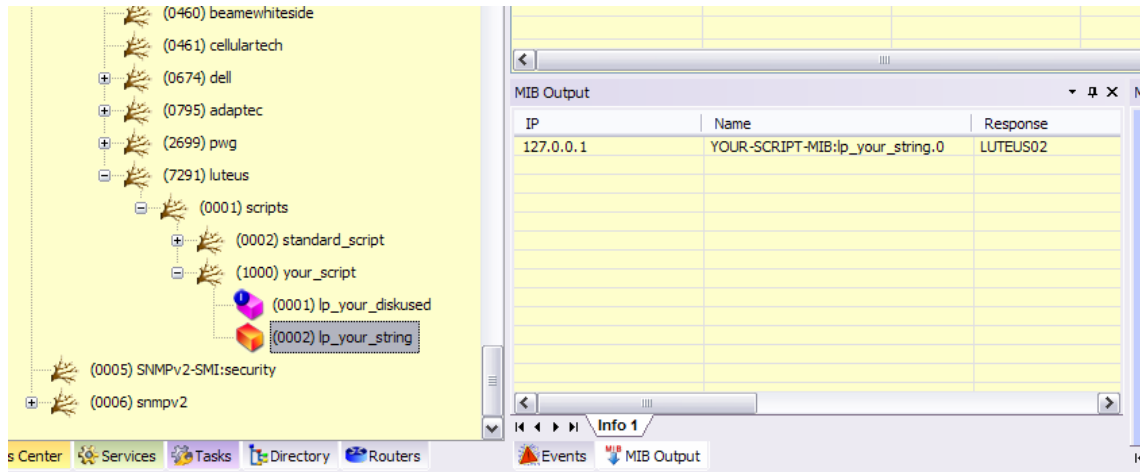
<LP_SCRIPT>

::= { your_script 2 }
```

Après compilation le fichier lp_your_string.lua est créé dans le répertoire bin/config/script.

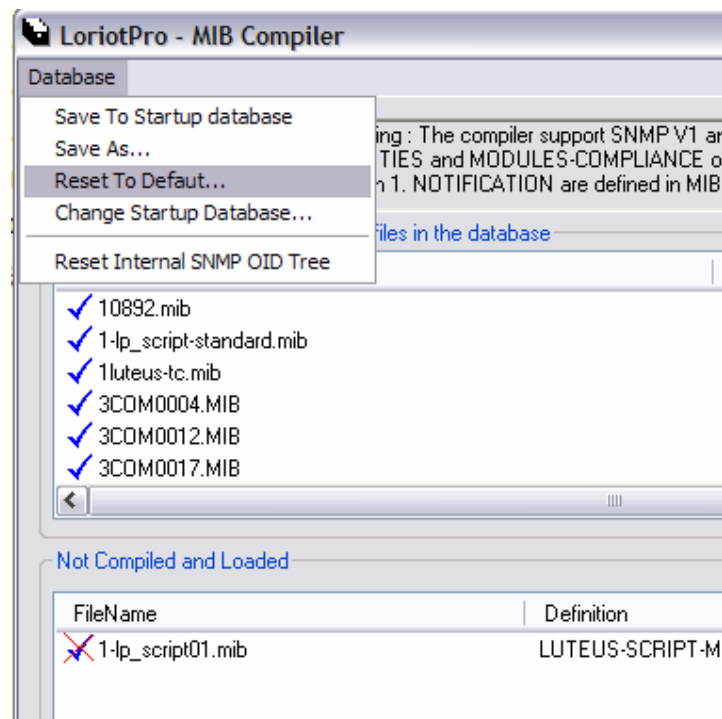


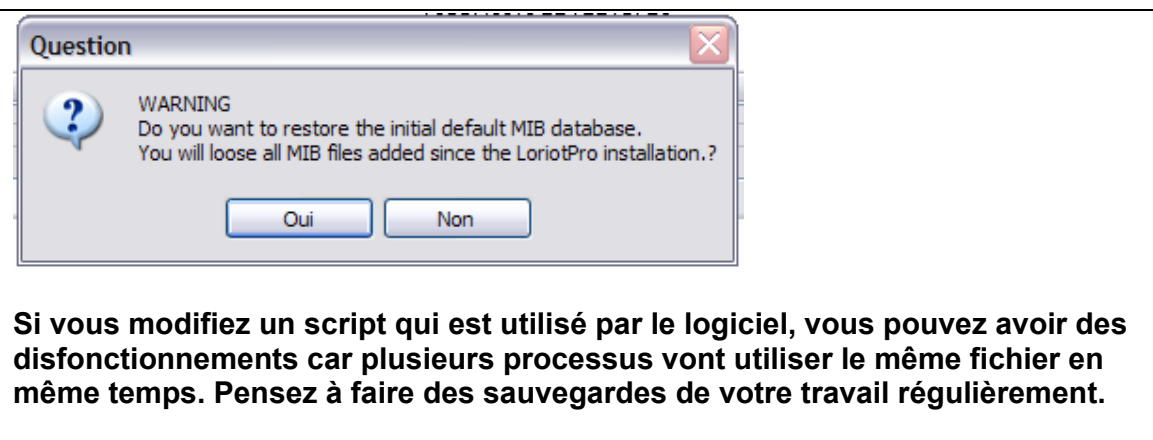
Le nouvel objet snmp virtuel est utilisable et renvoie une chaîne de caractères.



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, utiliser la fonction « Reset to Default... » du compilateur de MIB et relancer LoriotPro.

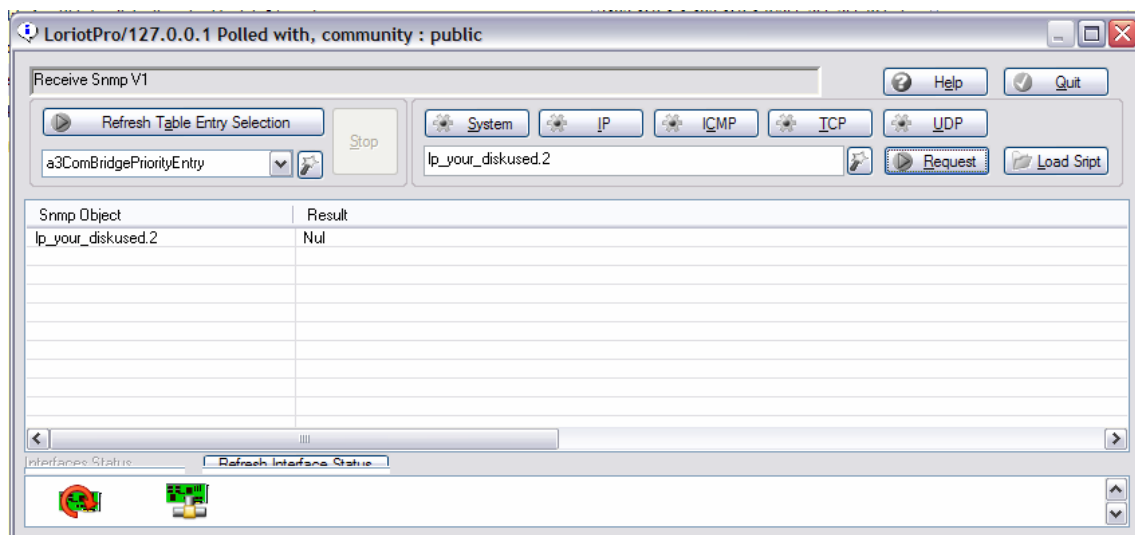
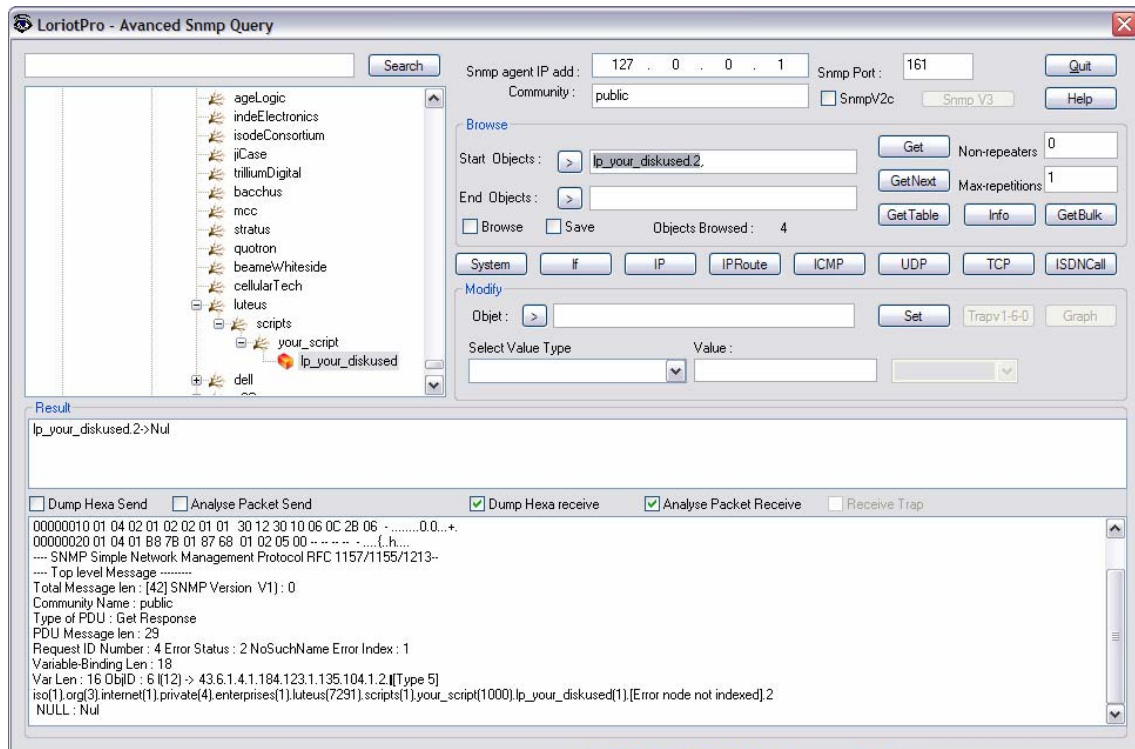




Si vous modifiez un script qui est utilisé par le logiciel, vous pouvez avoir des dysfonctionnements car plusieurs processus vont utiliser le même fichier en même temps. Pensez à faire des sauvegardes de votre travail régulièrement.

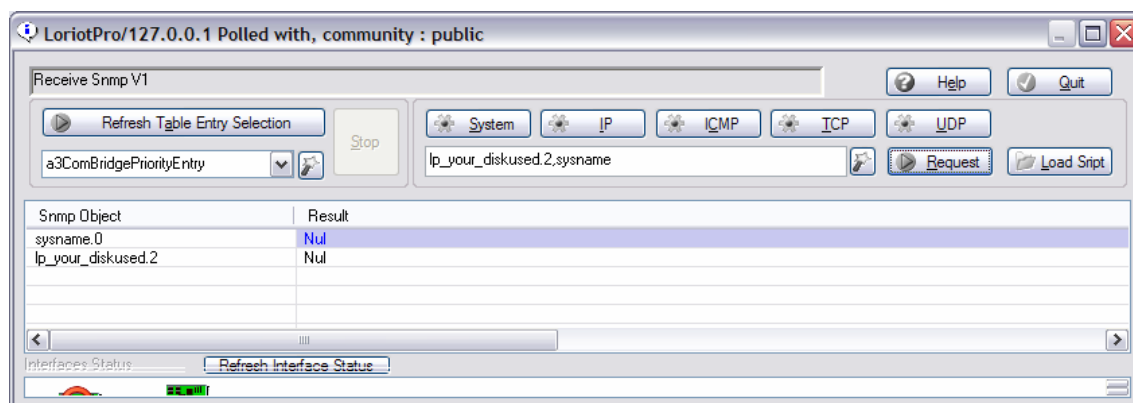
Exception d'utilisation des objets SNMP virtuel

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 plugin et module de LoriotPro travaillent avec des requêtes en mode synchrone et ils supportent les objets snmp virtuels.



ATTENTION

Mixé dans des requêtes avec des objet SNMP virtuel et SNMP réel 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*.

Utilisation des scripts LUA dans les ActiveView

Introduction

Les scripts utilisés par les ActiveView 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 un 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.

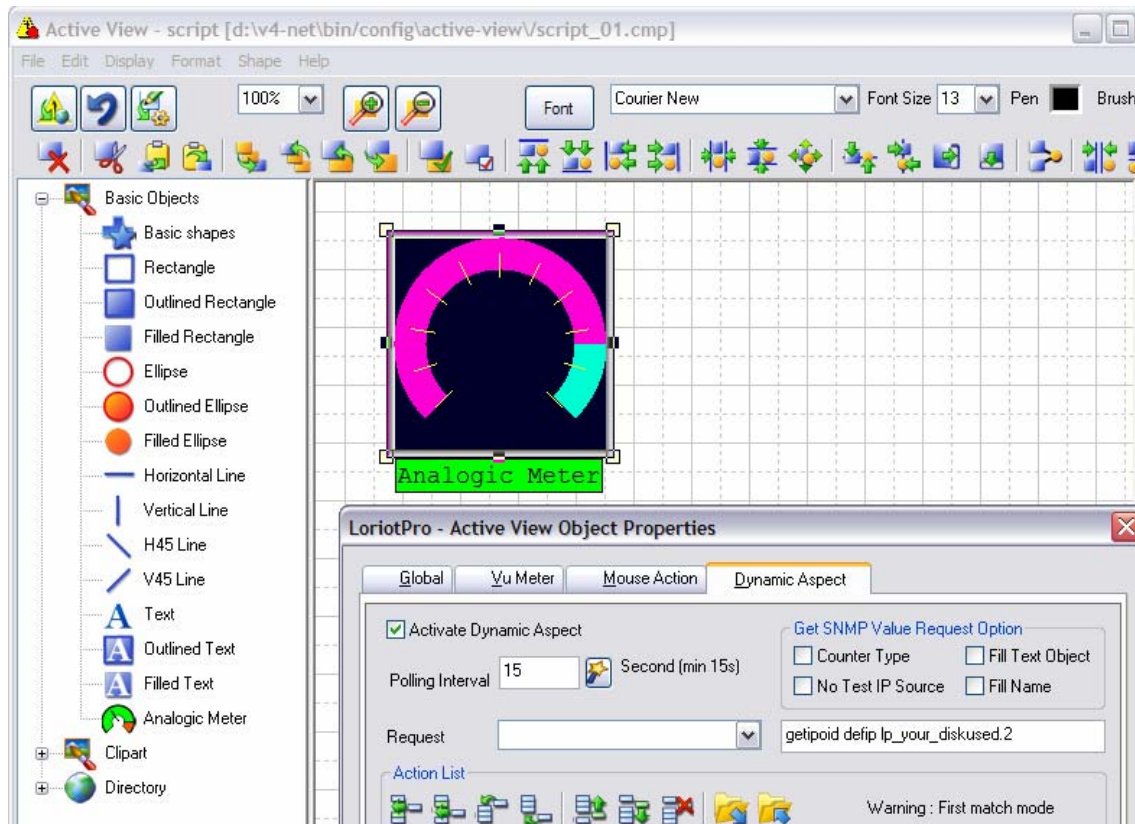
Type de scripts utilisable dans les ActiveView

Les ActiveView permettent un usage multiple des scripts LUA.

1. Un usage direct d'objet snmp virtuel est possible.
2. Un usage direct est possible en ligne de commande avec l'option Getipoid sous la forme d'une expression de type calculatrice 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 click 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

Usage direct d'objets SNMP virtuels

L'utilisation d'objet snmp virtuel ne pose aucun problème, ils sont utilisés comme des objets snmp standard.



Le mode « Wizard » permet de sélectionner les objets snmp virtuel, mais il ne supporte pas l'indexation Eventuelle de l'objet.

The screenshot displays the 'LoriotPro - MIB Object Description Browser Picker' window for IP [127.0.0.1]. The main tree view lists various MIB objects, with 'lp_your_diskused' selected under the 'scripts' folder. The right-hand pane provides details for the selected object, including its name, children, MIB description, and syntax. Below this, a 'Description' field contains a Lua script snippet: `"Test un calcul sur = (lp.Get(lp_host,'hrstorageused.index') / lp.C`. In the background, an 'Analogic Meter' is visible, and the 'LoriotPro - Active View Object Properties' dialog is open, showing a 'Request' field with the value 'getipoid defip lp_your_diskused.2' and an 'Action List' table.

Condition	Value	Set C...	Set Event	Set Threshold	IP reference
00 (=) equal	#ERROR	0	1	0.0.0.0	0.
01 (>) greater than	95	0	1	0.0.0.0	0.
08 Any	(null)	0	1	0.0.0.0	0.

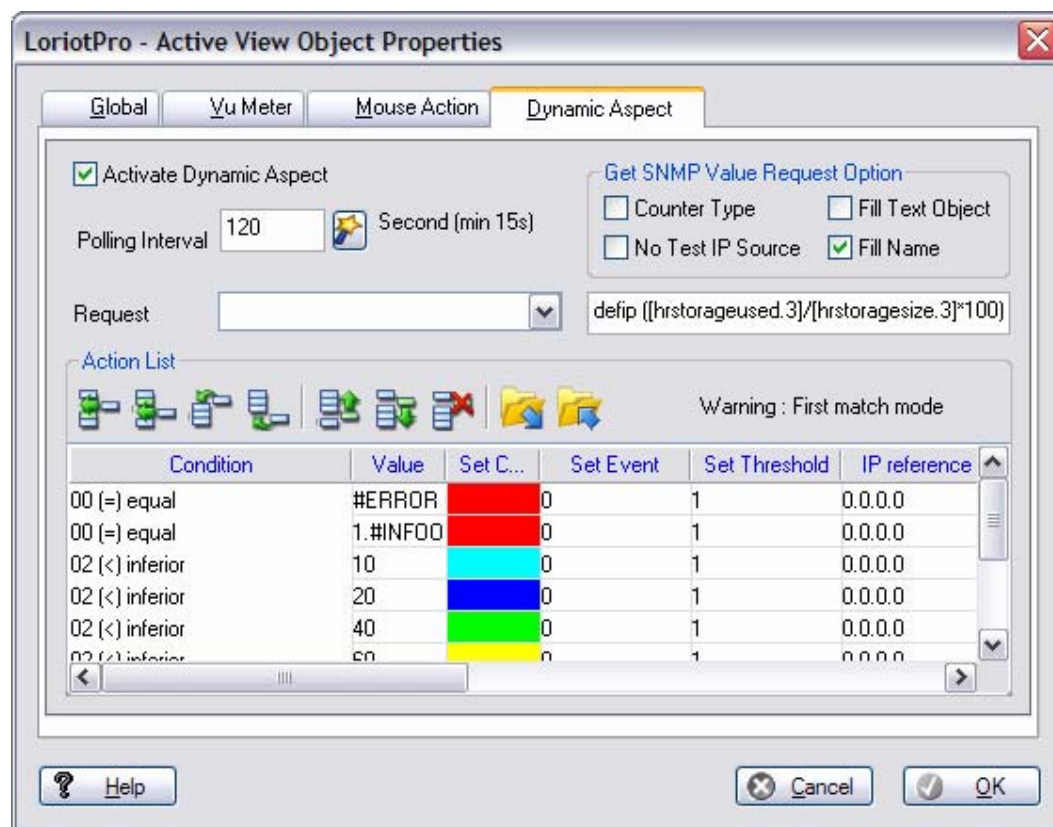
Calculatrice scientifique

En utilisant la fonction GetipOld standard il est possible de réaliser des opérations mathématiques simples sur une collecte d'objet 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 calculatrice scientifique (langage C, C++) avec des parenthèses. Pour la syntaxe des fonctions mathématiques avancé 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-resource-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)*

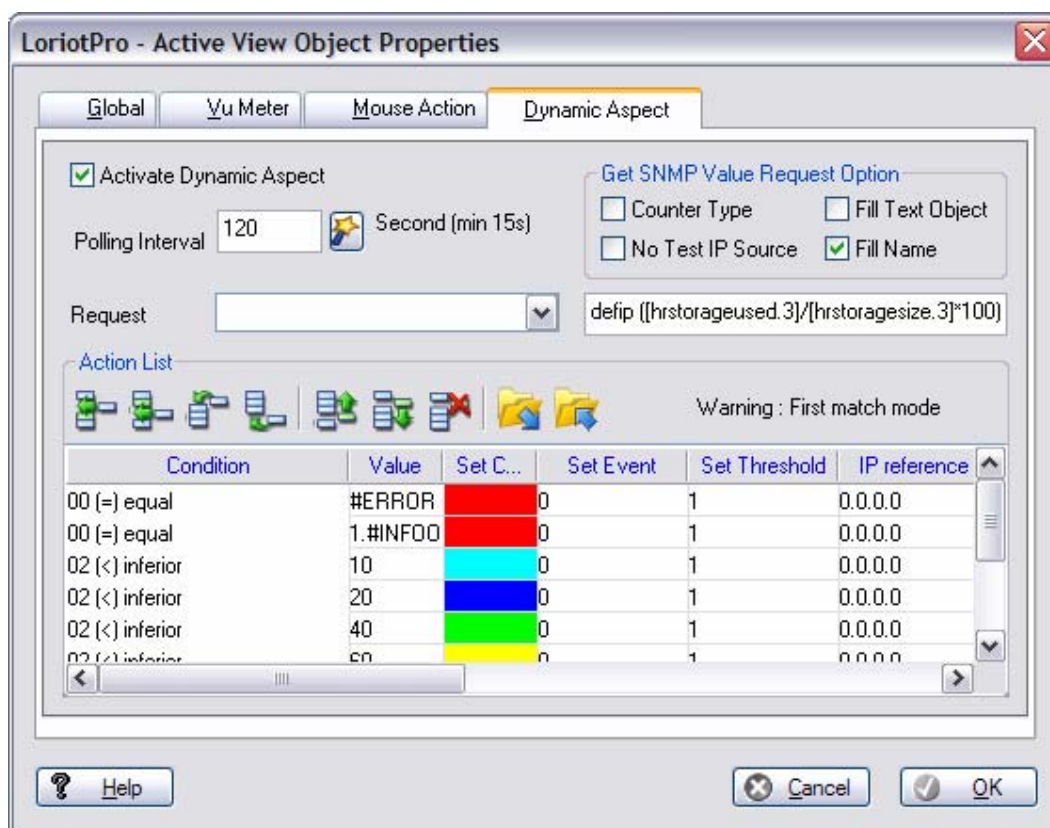
Non souhaité mais ça fonctionne (l'analyseur syntaxique (parser) de l'ActiveView supporte 50 blocs maximum)

*(([hrstorageused.3] / [hrstoragesize.3] * 100) / 1000)*

Valeur possible en retour d'expression

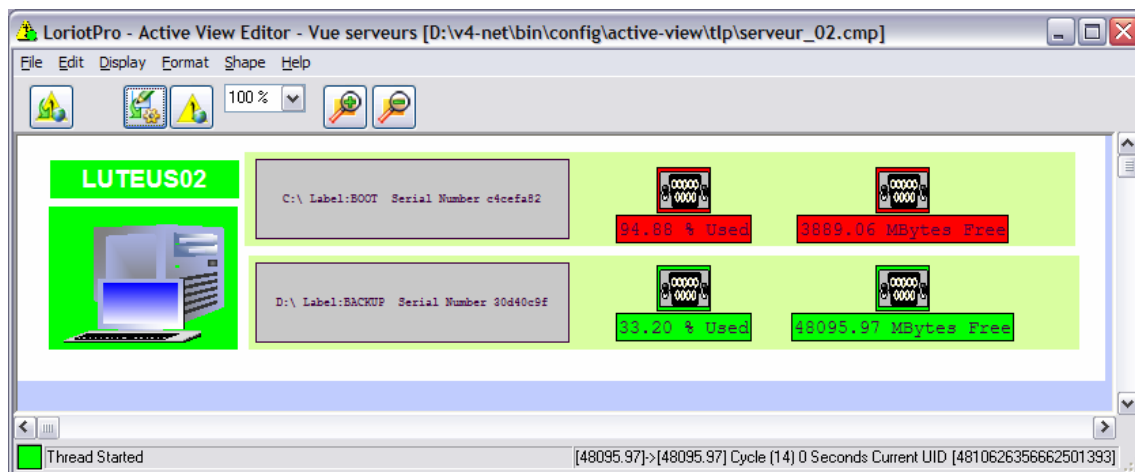
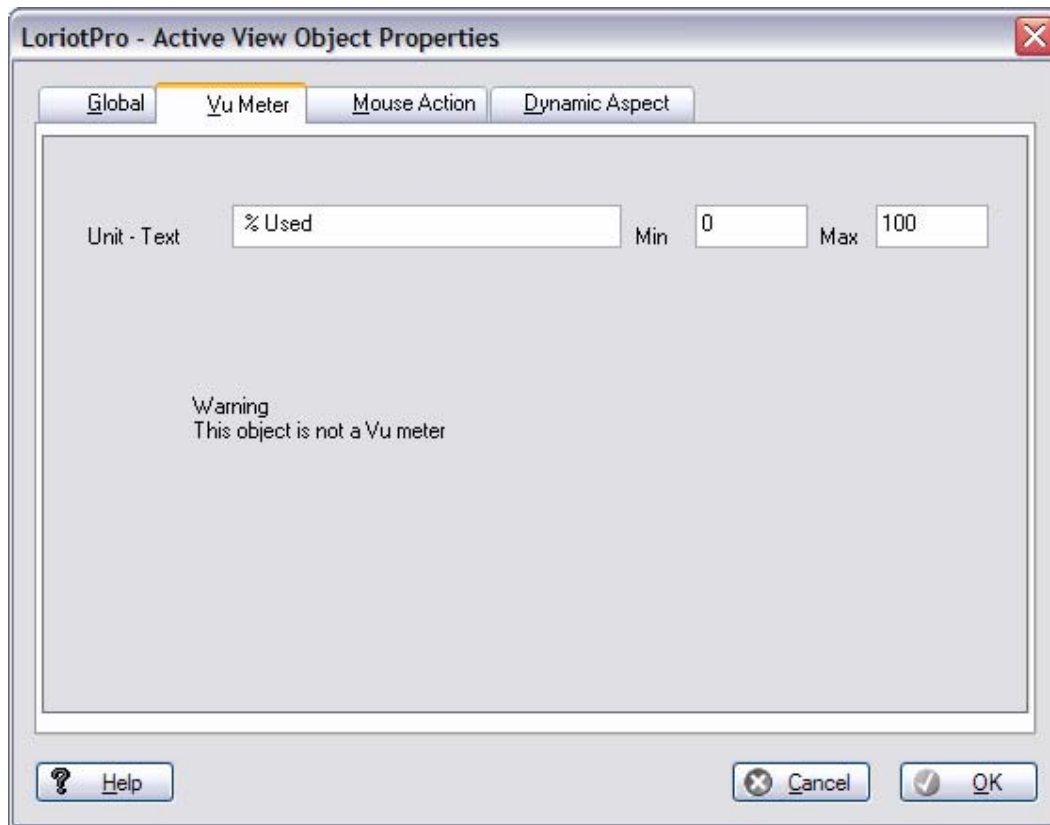
#ERROR si la syntaxe est incorrecte ou si les objets snmp (OID) ne sont pas trouvés

1.#INF00 si il y a une erreur de division par zéro

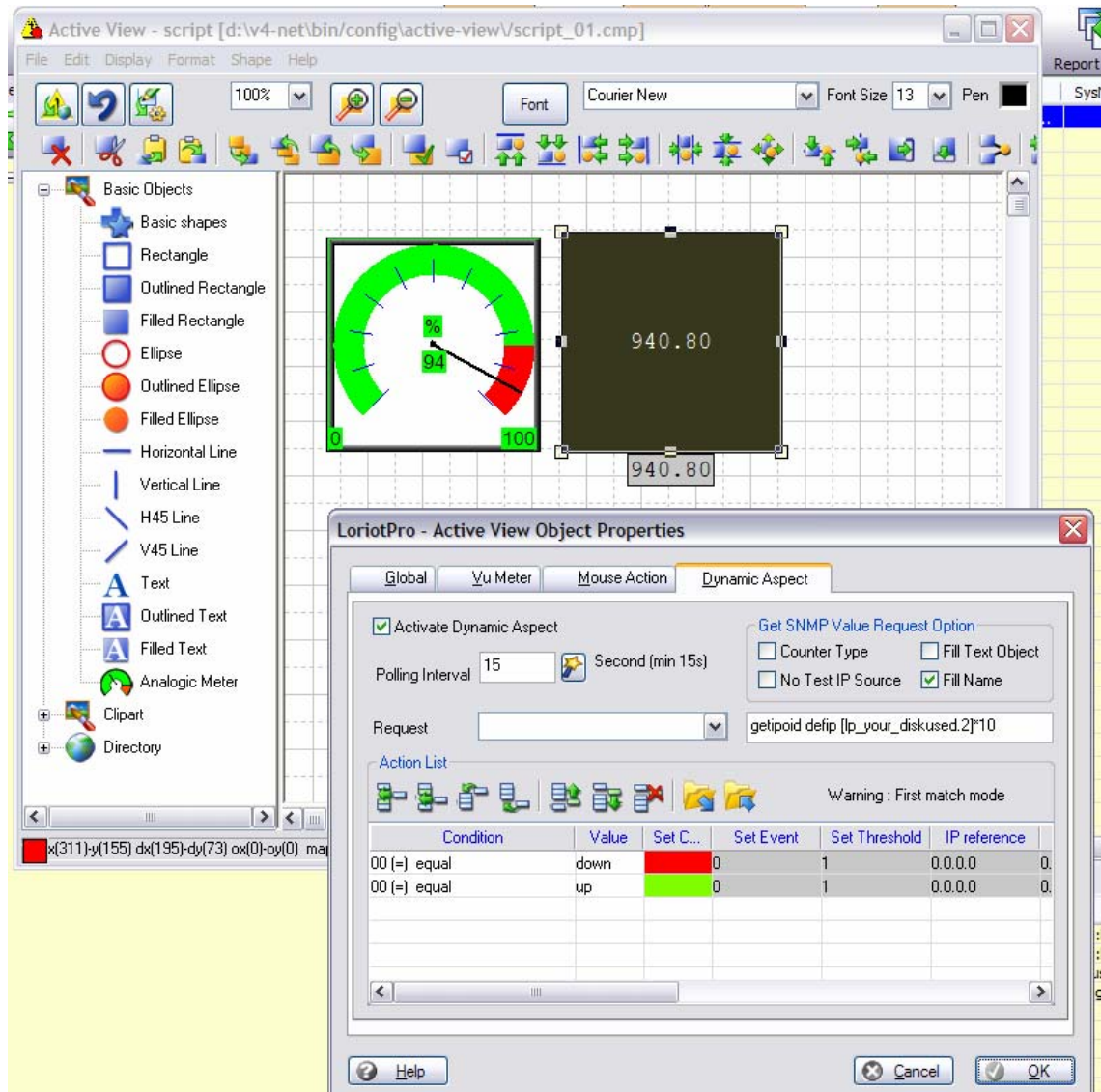


Si on mets une valeur dans unit elle est automatiquement ajoutée au nom (« name ») de l'objet avec l'option 'fill name' cela permet d'avoir des noms dynamiques avec une unité.

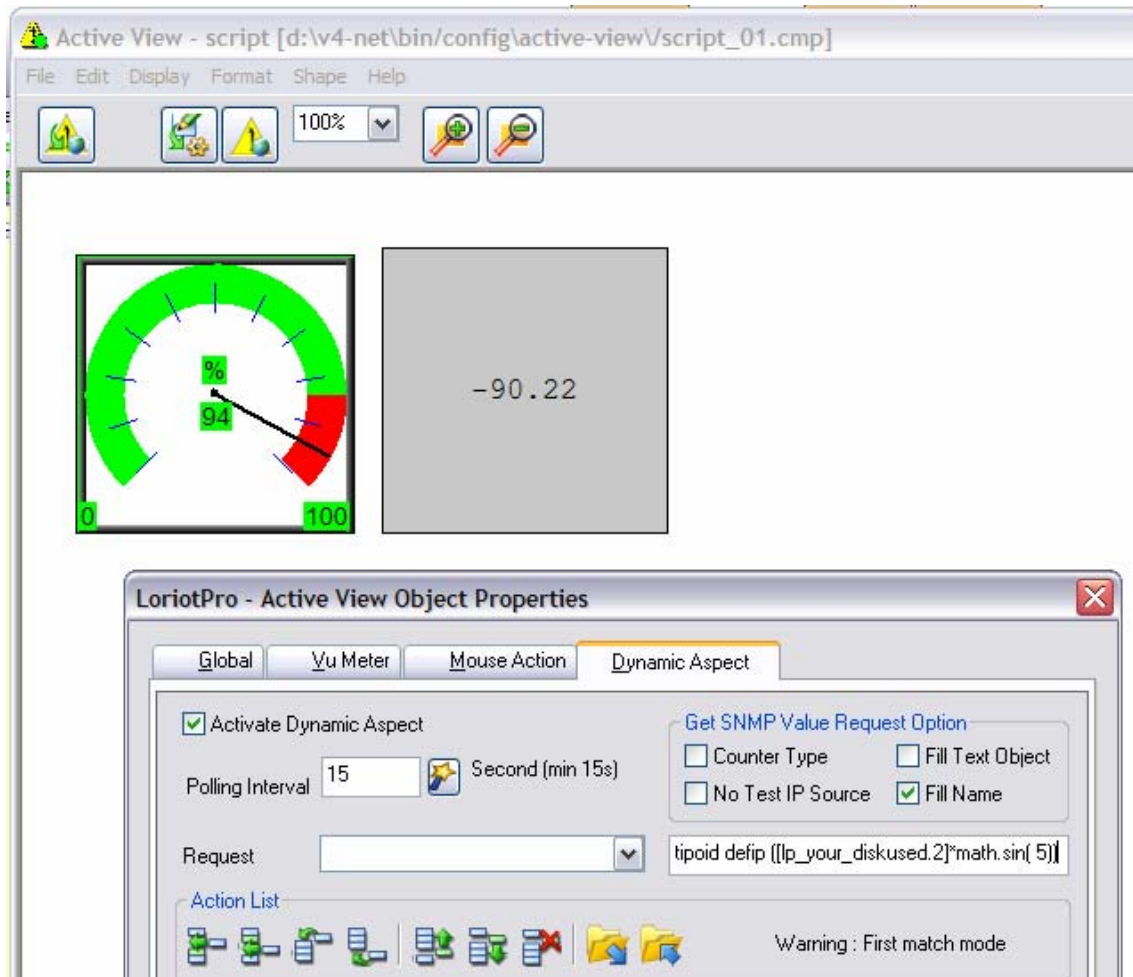
L'option : 'fill name' permet de remplacer le nom de l'objet par le résultat.
L'option unit dans (vu meter) est ajoutée au nom si il existe.



Dans l'exemple suivant on utilise un objet snmp virtuel pour réaliser le calcul.



Cet exemple intègre du calcul avec des sinus.



Extrais 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.

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'un 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

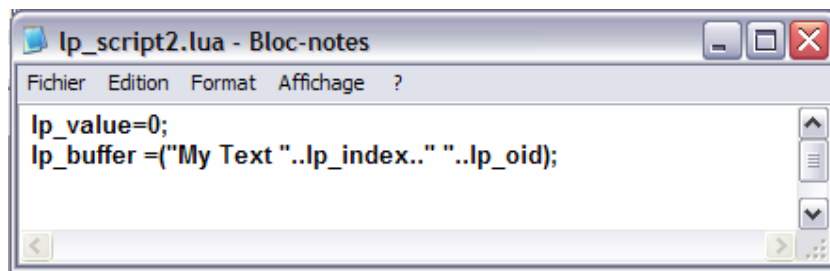
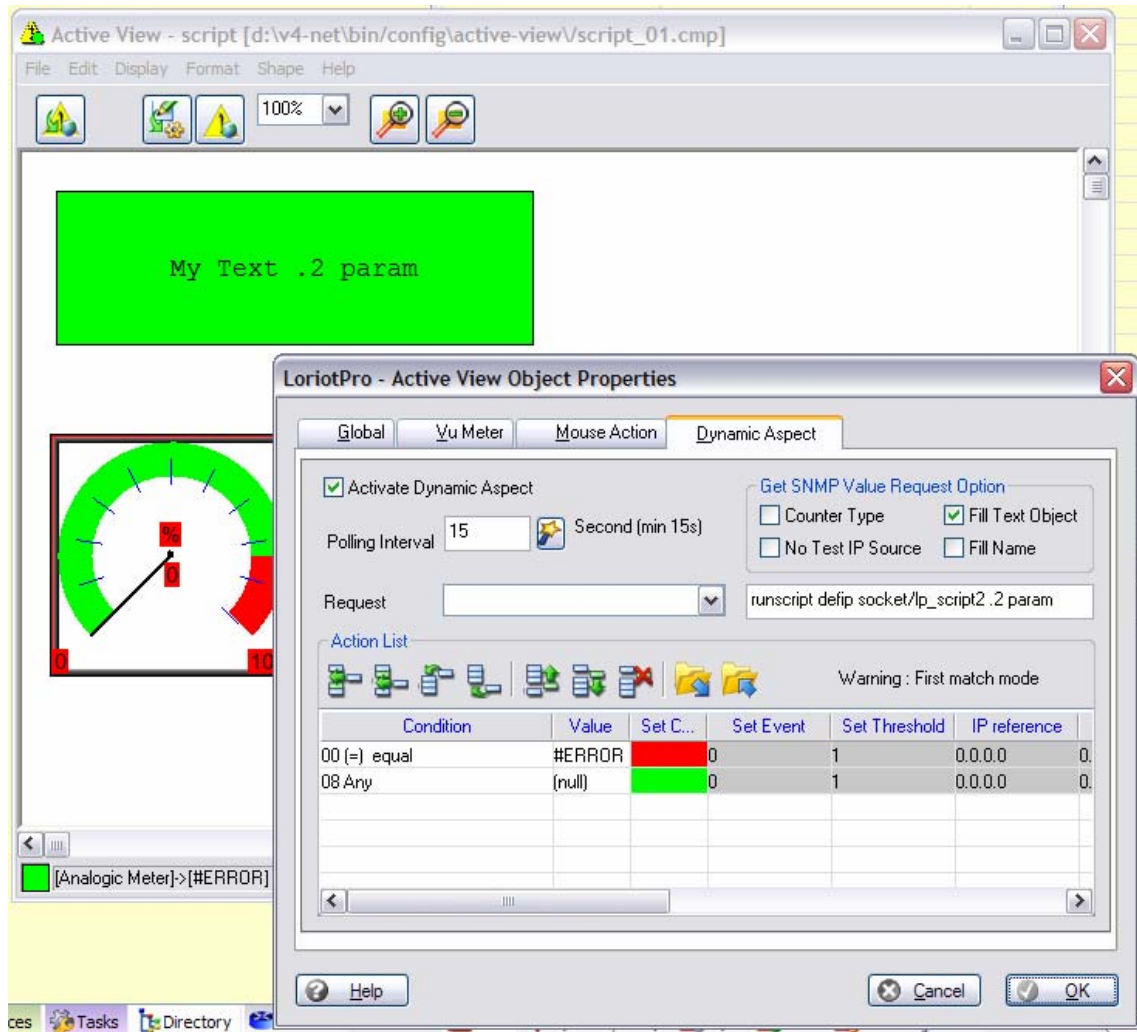
Le script sera exécuté à chaque polling intervalle. Si vous voulez exécuter un script de façon unitaire, utilisez les fonctions de menu contextuel ou faire un double click de la souris associé à l'objet graphique.

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 du Host 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 Ip_host 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 Exemples Soit le fichier script1.lua dans <i>bin/config/script</i> . RunScript defip script1 Soit le fichier script1.lua dans <i>bin/config/script/socket</i> . RunScript defip socket/script1
Ip_index	Ce paramètre est optionnel, le programme passe ce paramètre dans la variable Ip_index du script.

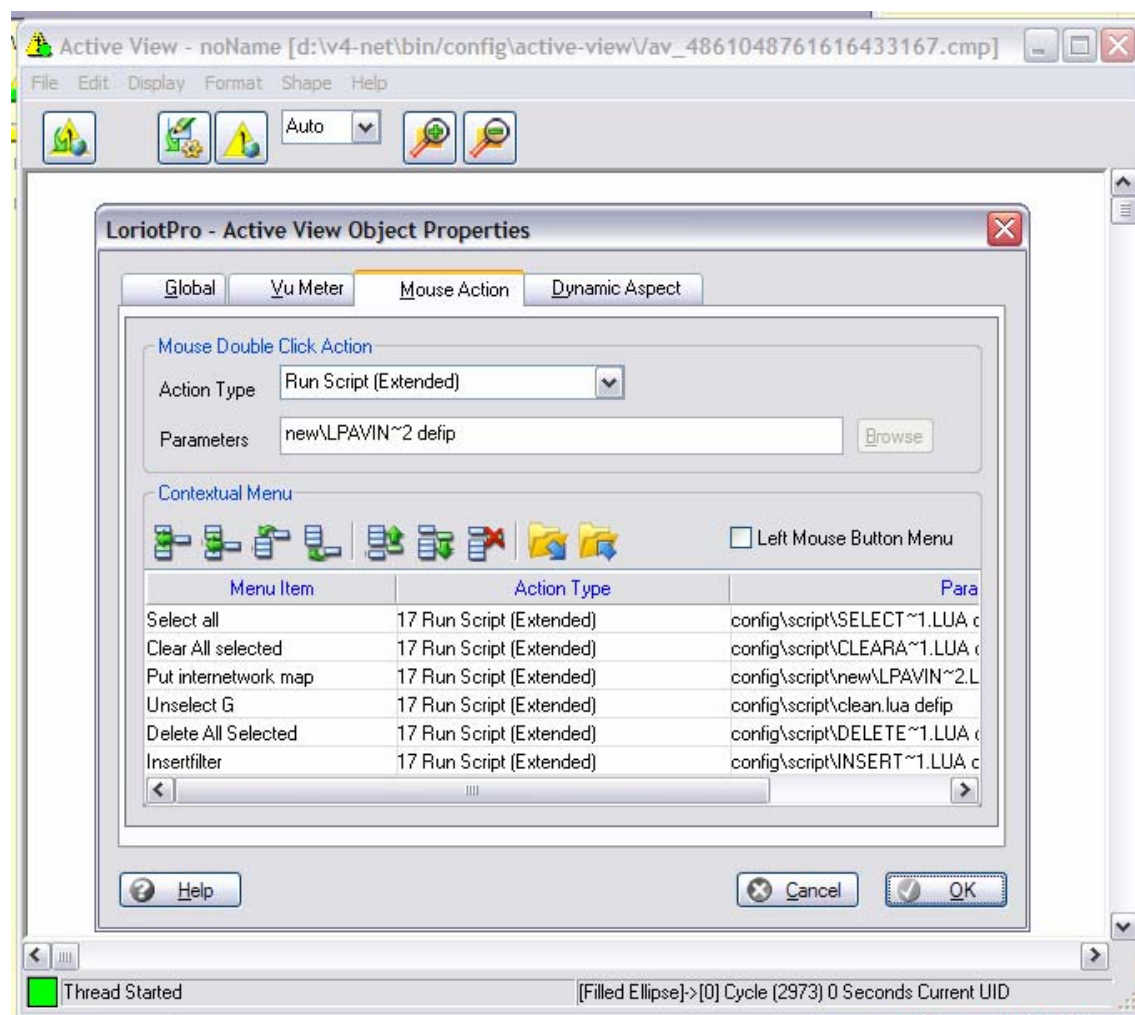
	<p>Attention</p> <p>Ce paramètre ne doit pas contenir de caractères blancs.</p> <p>Exemples</p> <p>Soit le fichier script1.lua dans <i>bin/config/script</i>.</p> <pre>RunScript defip script1 .2</pre> <p>Soit le fichier script1.lua dans <i>bin/config/script/socket</i>.</p> <pre>RunScript defip socket/script1 .2</pre>
<p>lp_oid</p>	<p>Ce paramètre est optionnel le programme passe ce paramètre dans la variable lp_oid du script.</p> <p>Attention</p> <p>Ce paramètre ne doit pas contenir de caractères blancs et pour l'utiliser il faut un paramètre lp_index dans la ligne même si il ne sert à rien.</p> <p>Exemples</p> <p>Soit le fichier script1.lua dans <i>bin/config/script</i>.</p> <pre>RunScript defip script1 .2 param</pre> <p>Soit le fichier script1.lua dans <i>bin/config/script/socket</i>.</p> <pre>RunScript defip socket/script2 .2 param</pre>



socket/lp_script2

Script utilisé avec la fonction DoubleClick

Un script peut être associé à l'action de double click sur un objet graphique. Ceci permet de dérouler un script sur action uniquement et non plus de façon régulière (polling) et continue.

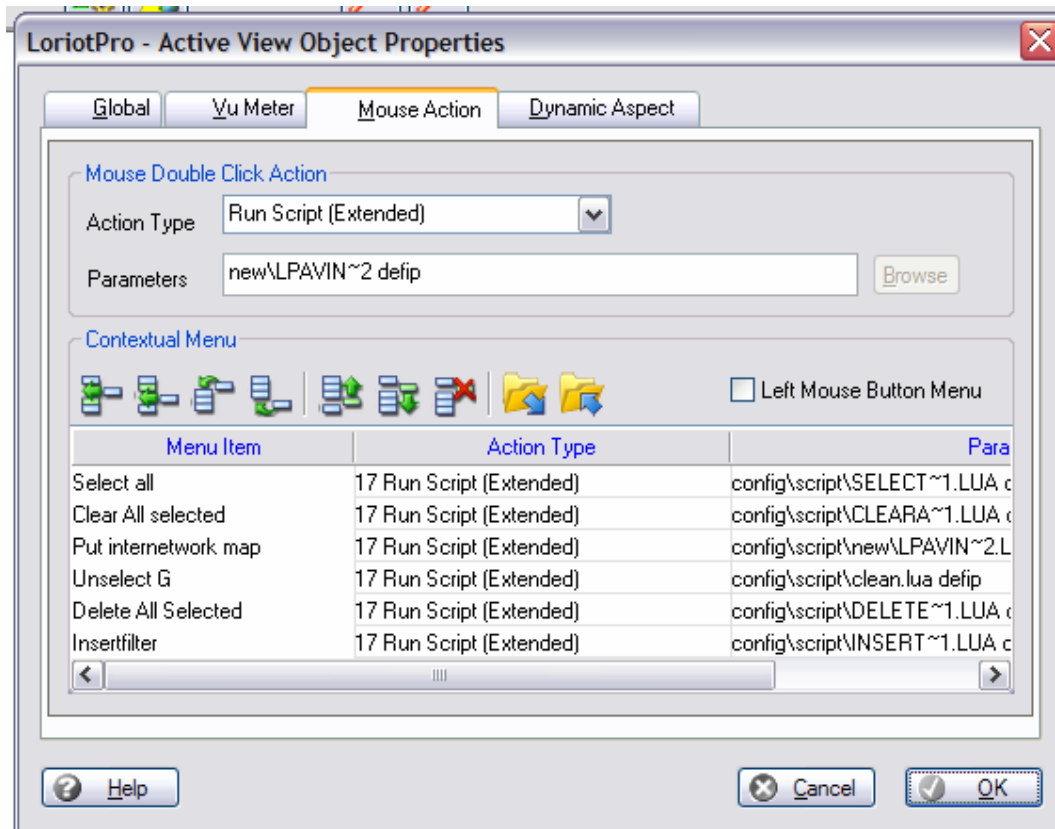


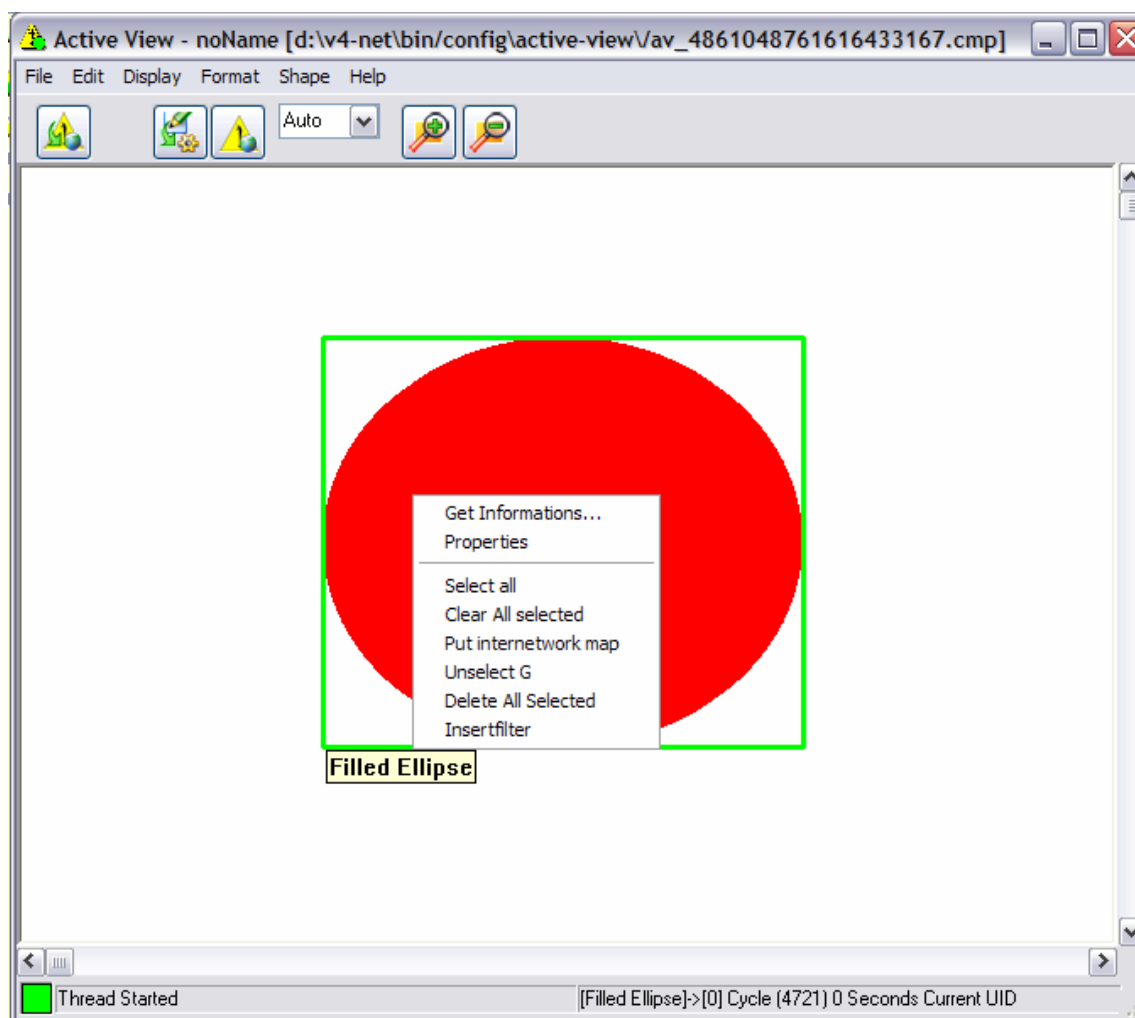
La syntaxe complète des paramètres est :

filename.lua ip_adresse index oid

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.

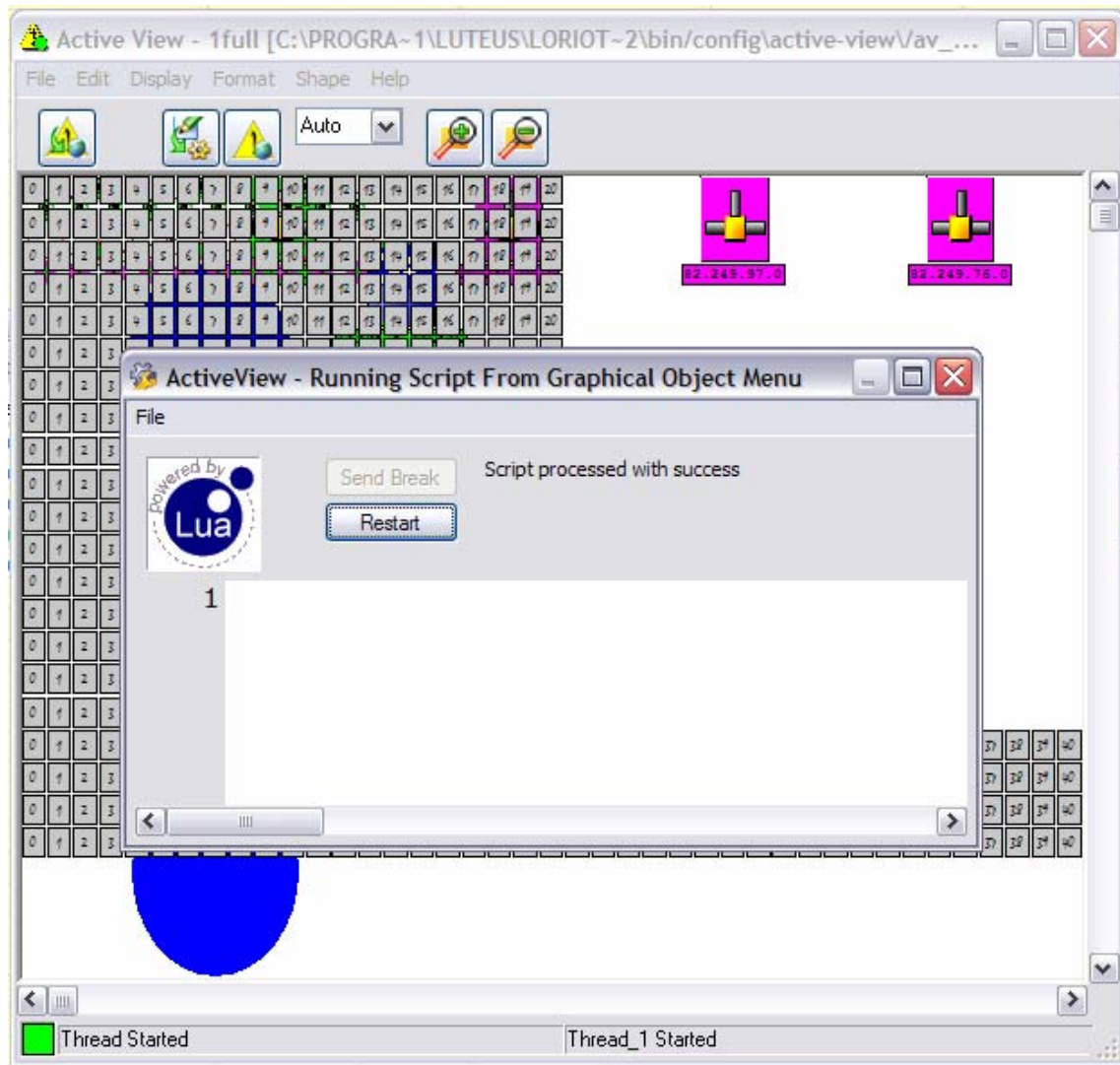




La syntaxe complète des paramètres est :

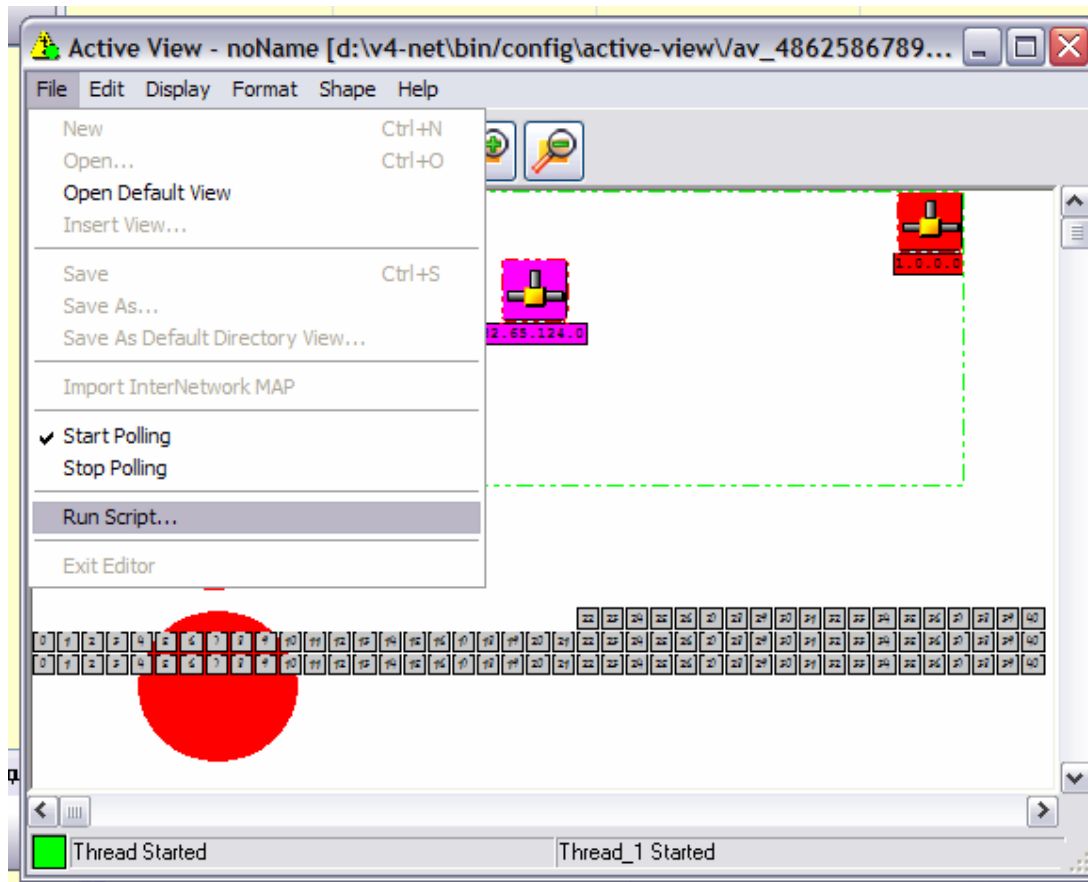
filename.lua ip_adresse index oid

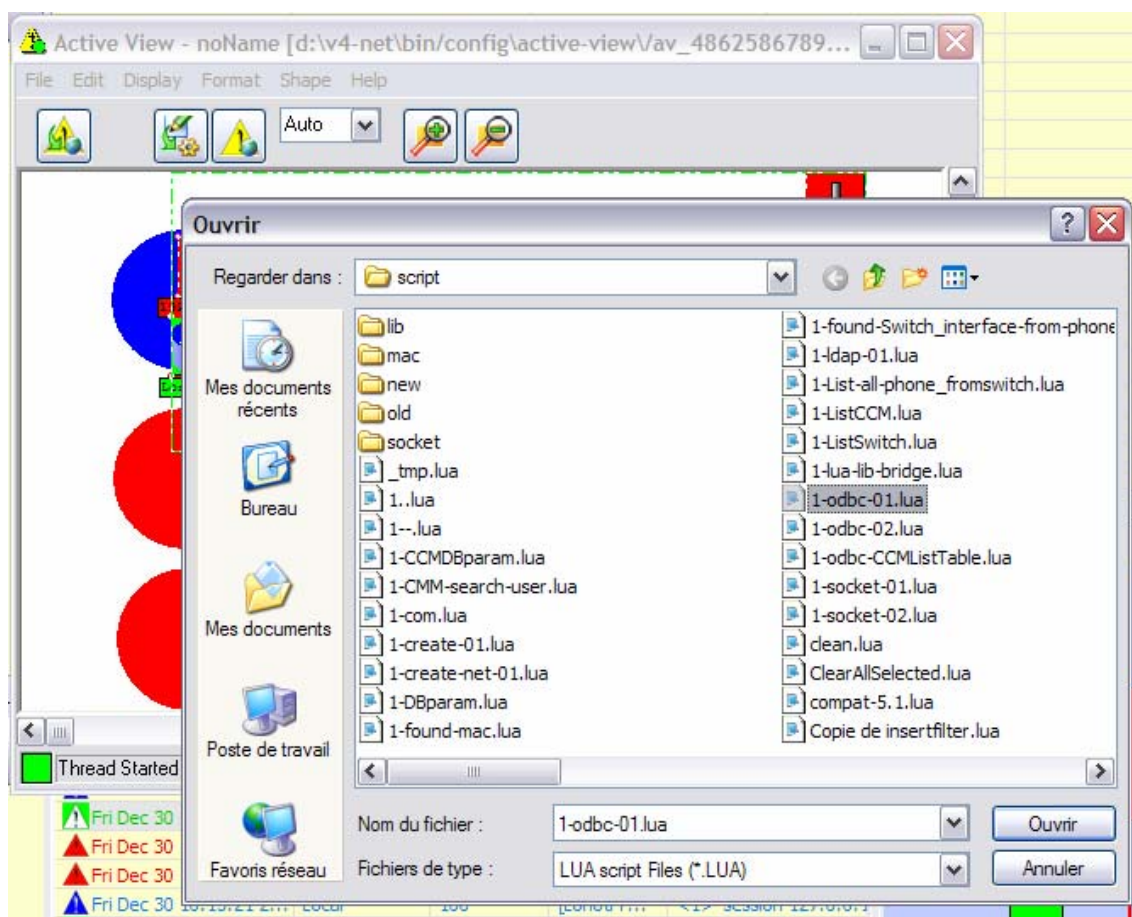
Une fenêtre pour gérer l'exécution du script est affichée.

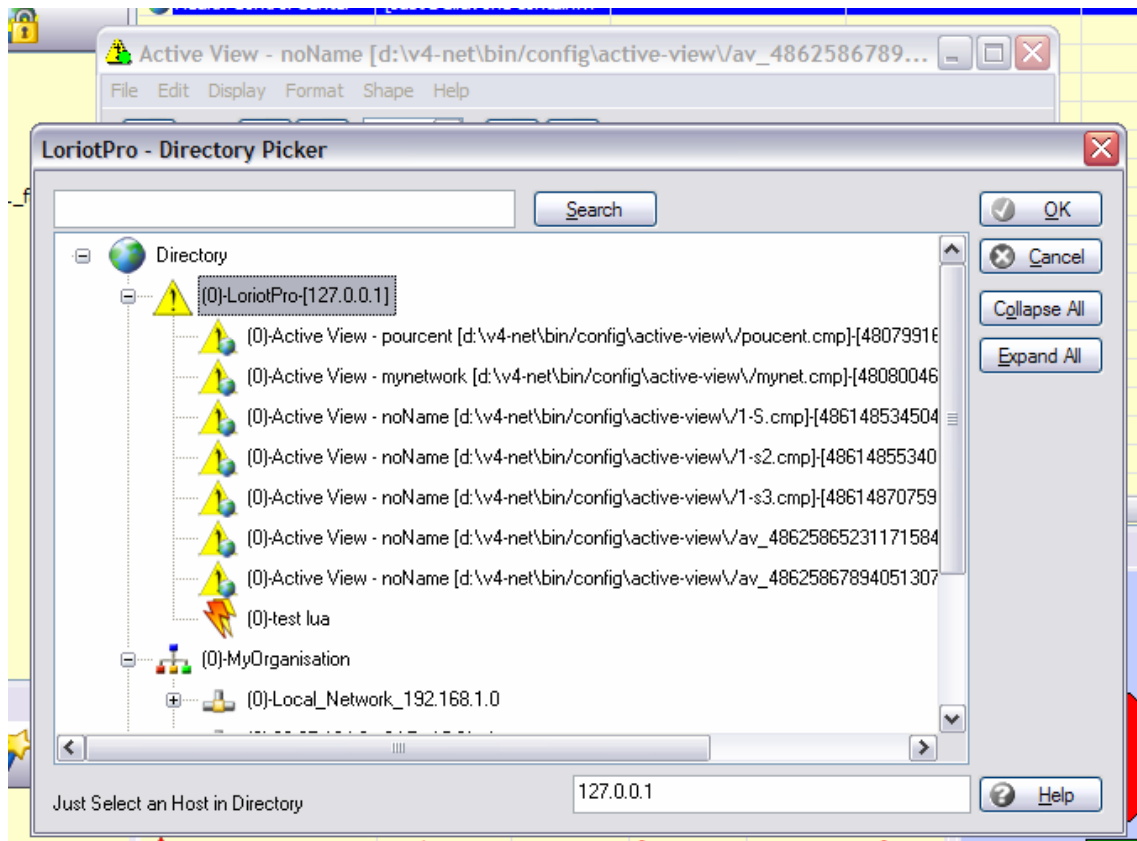
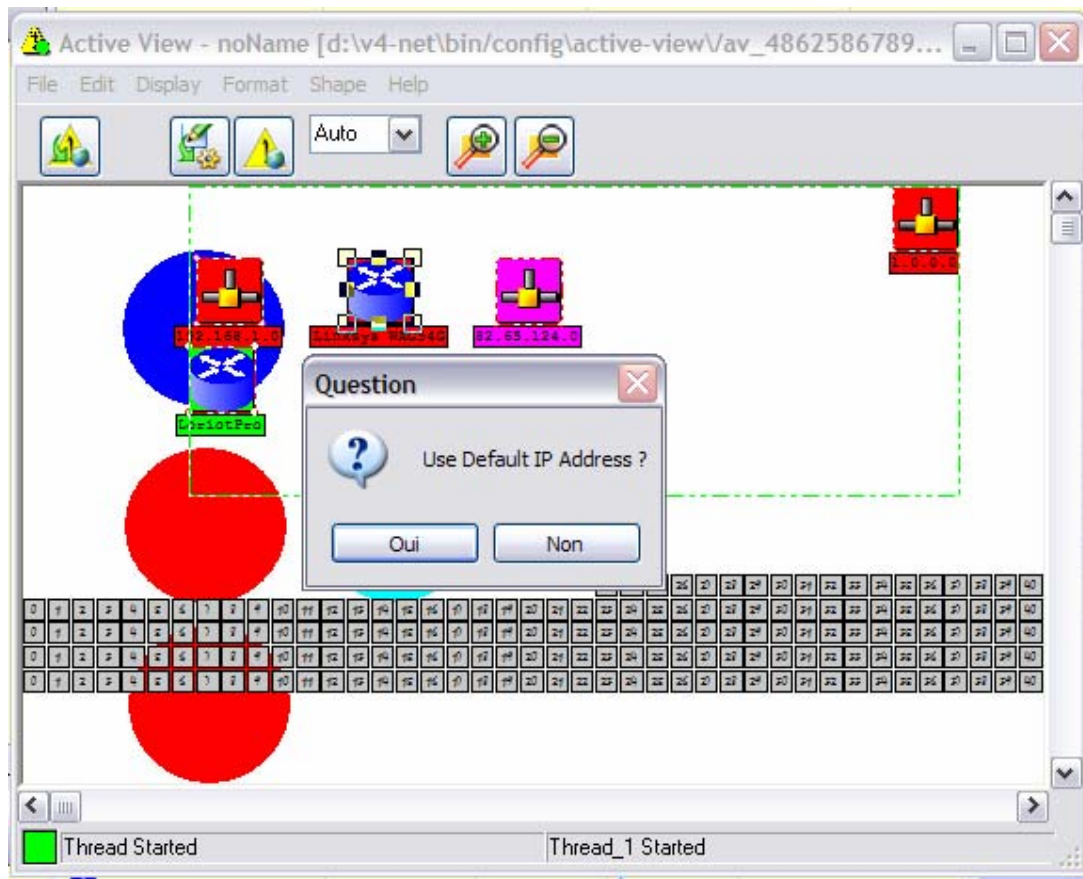


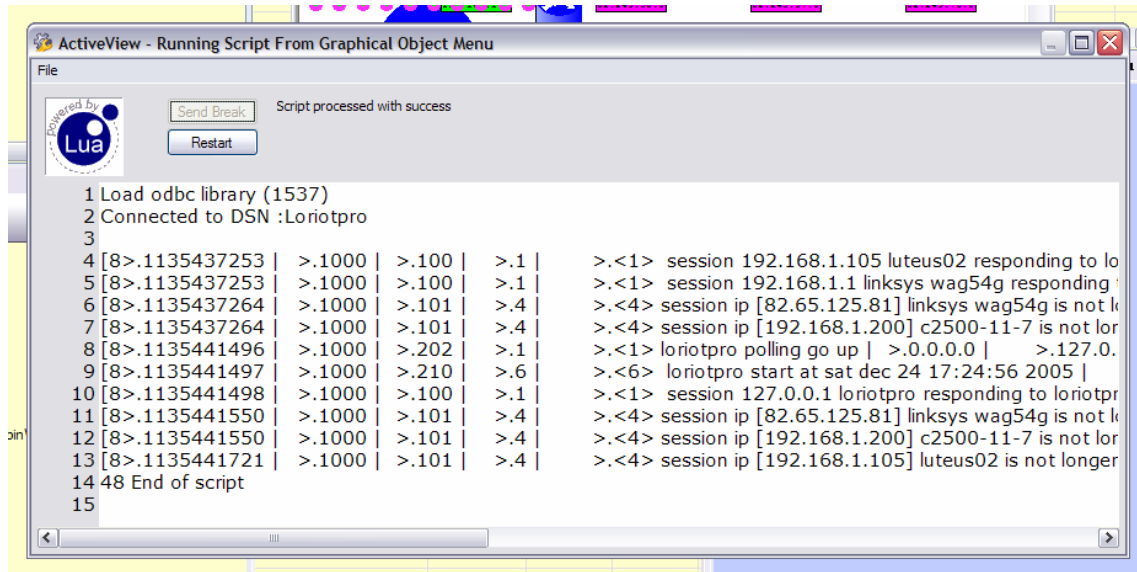
Cliquer sur l'icône croix rouge pour quitter cette fenêtre.

Utilisation directe de script dans une ActiveView



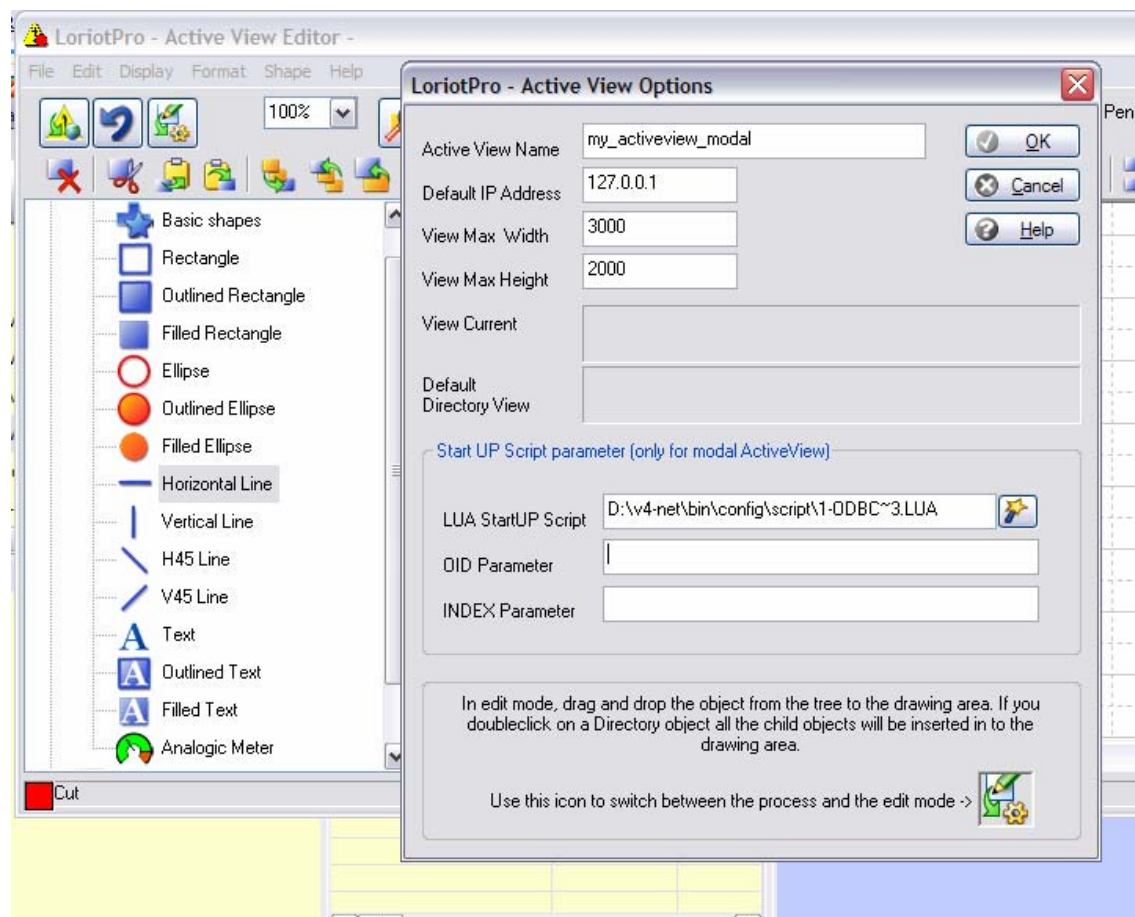






Utilisation de Script d'initialisation pour les ActiveView Modal

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 click de souris sur un objet graphique d'ActiveView.

Parametres	Description
Default IP Adress	Le parametre Ip_host passé au script. Attention Dans le cas d'une utilisation à travers un shortcut de directory le Ip_host passé au script est celui indiqué dans la ligne de commande du shortcut (cela permet de travailler avec des ActiveView utilisant la notion de « defip ») et ce paramètre est remplacé par l'adresse IP du shortcut.
LUA StartUP Script	Le fichier lua lancé
OID Parameter	Le parametre Ip_oid passé au script

INDEX Parameter	Le parametre Ip_index passé au script
-----------------	--

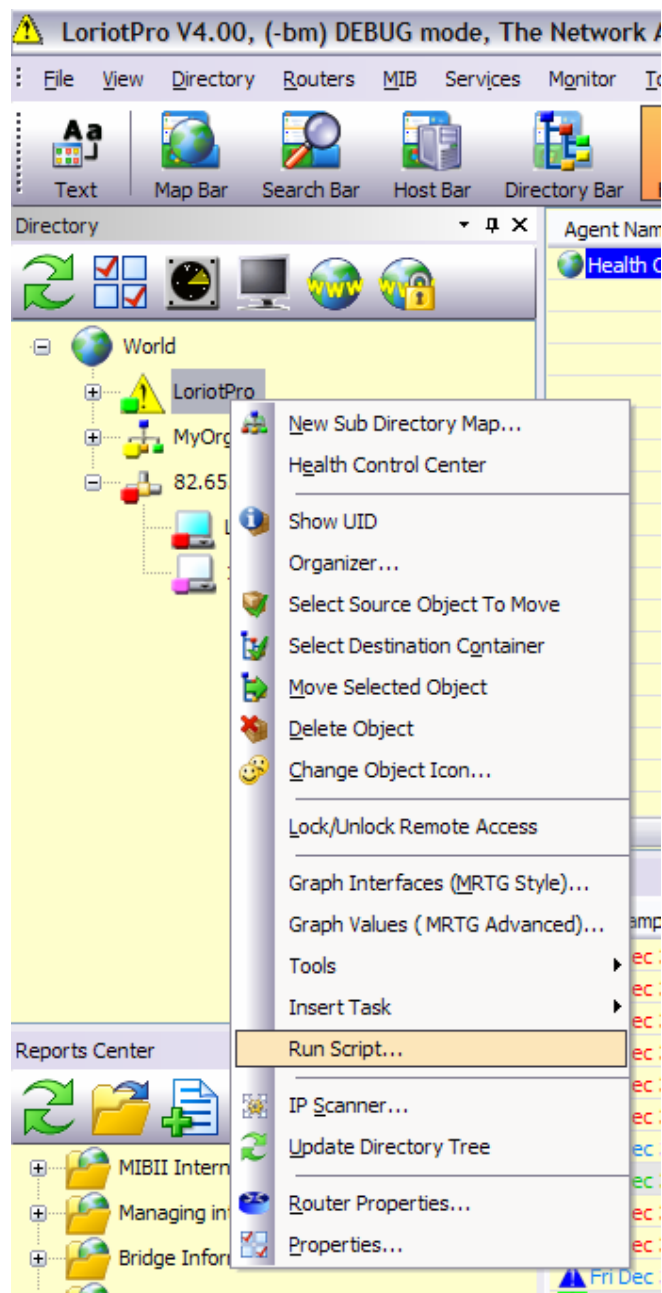
Reportez vous au chapitre « ShortCut associant une ActiveView Modal et un Script LUA » pour une description de l'utilisation de cette fonction.

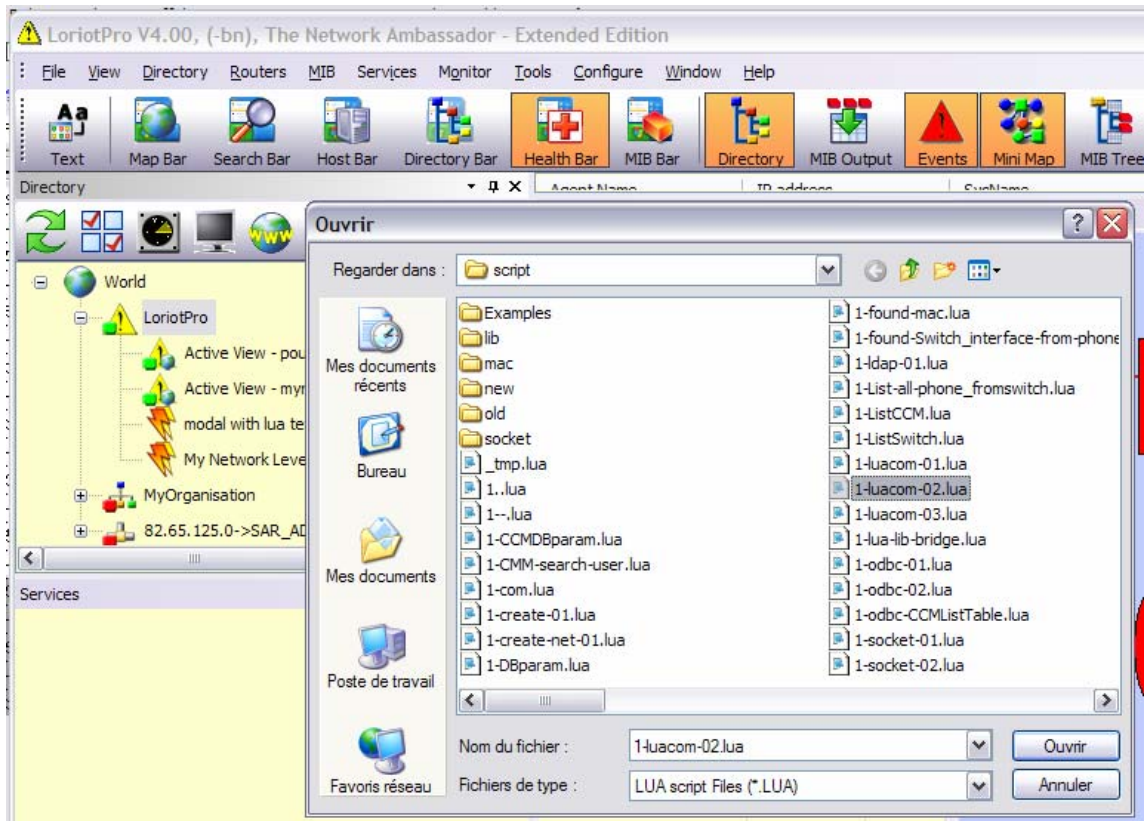
Utilisation direct de script

A partir des menus du logiciel

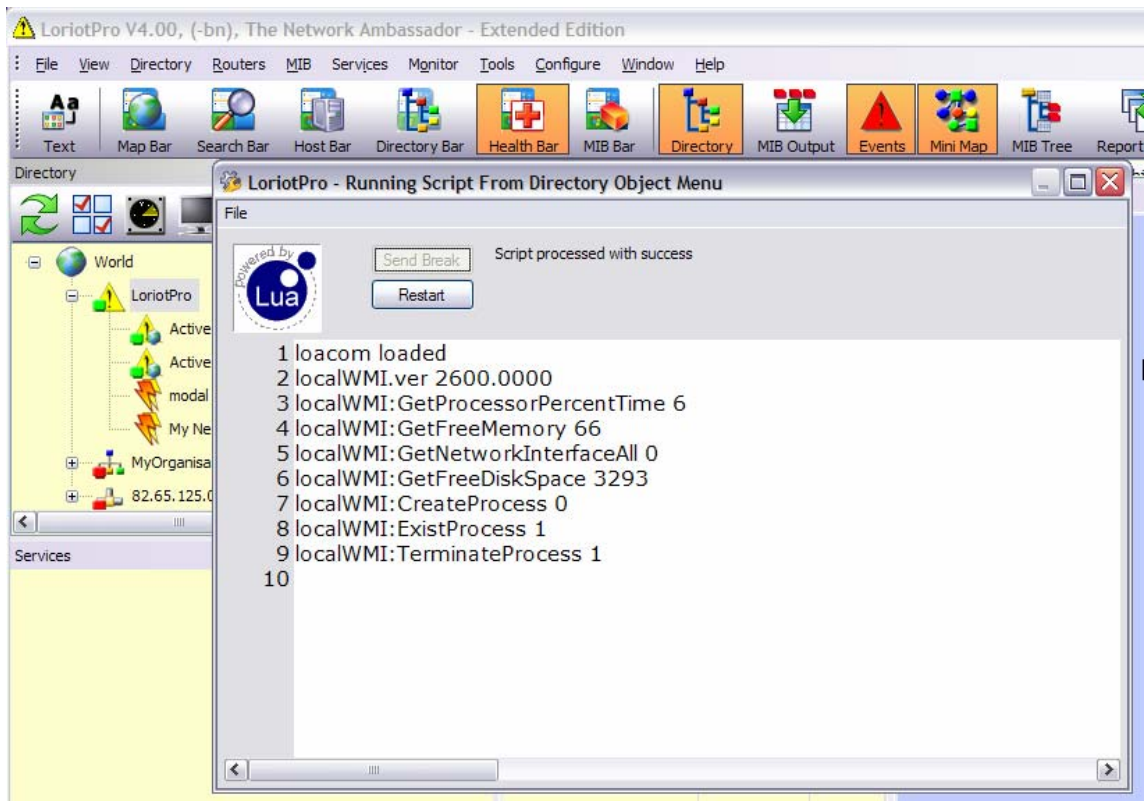
Il est possible de lancer un script à partir des menus du logiciel. Le logiciel passe dans la variable **Ip_host** l'adresse IP du Host par défaut sélectionné dans la directory

Menu contextuel d'un host

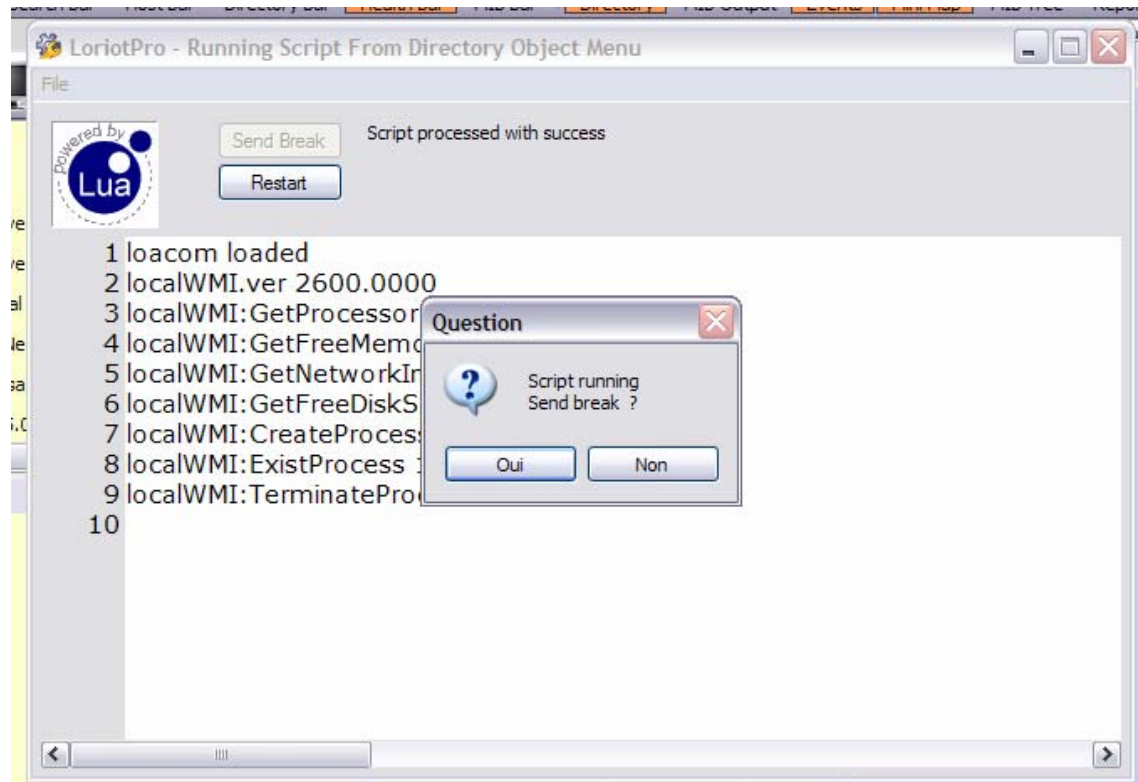




Sélectionner votre script.



La fenêtre d'exécution des Script 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 attendre la fin de l'exécution du script.



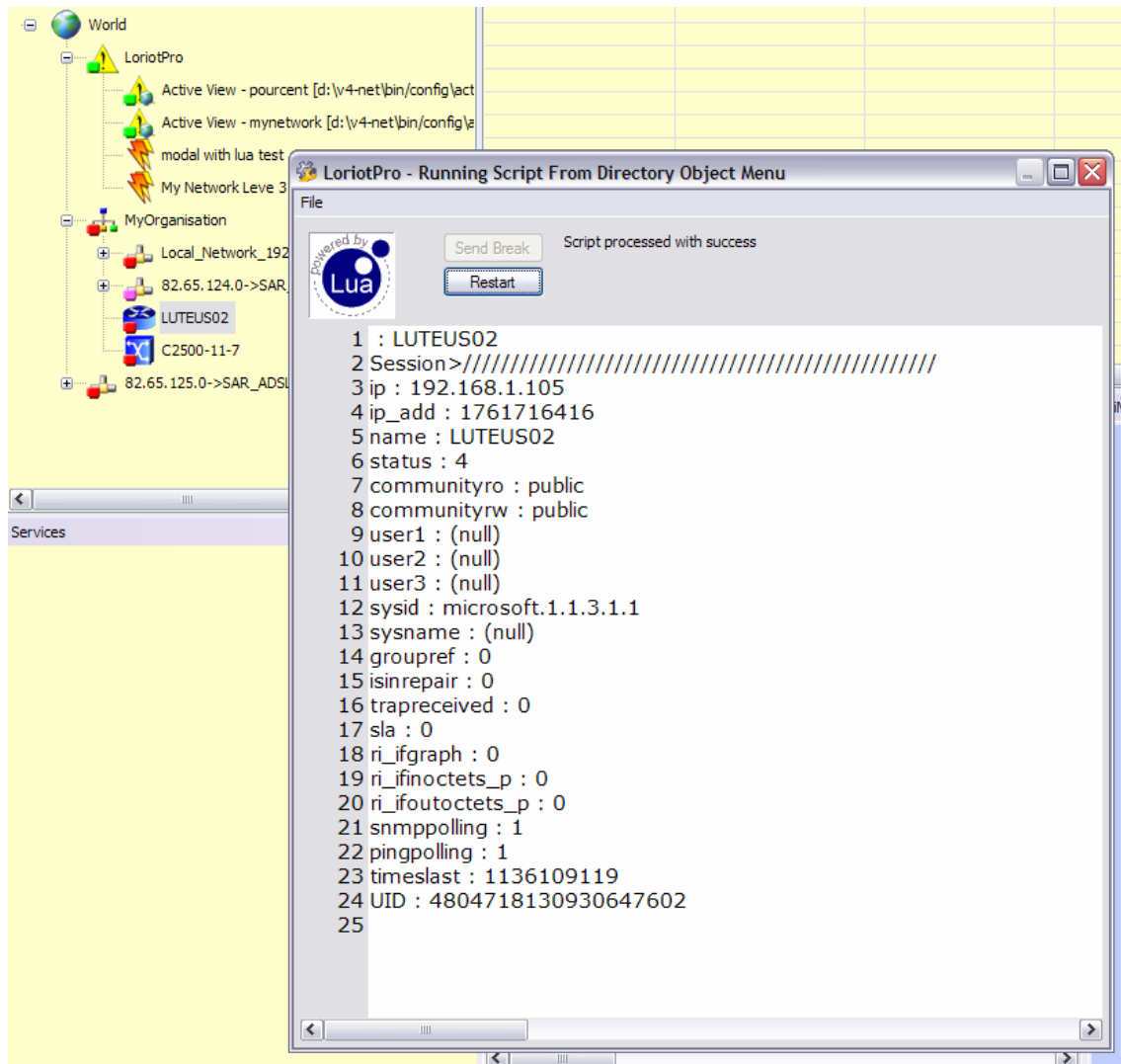
Dans cet exemple, on vérifie que la variable **Ip_host** correspondant à l'adresse IP du Host sélectionné dans la directory est bien passé au script :

```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\1-ip_info.lua
Files Edit View Insert Search Options Compiler Value
11  lp_value=lp.GetIPInformation(lp_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_ifinotets_p : ",aa["ri_ifinotets_p"], "\n");
37  lp.Print("ri_ifoutotets_p : ",aa["ri_ifoutotets_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

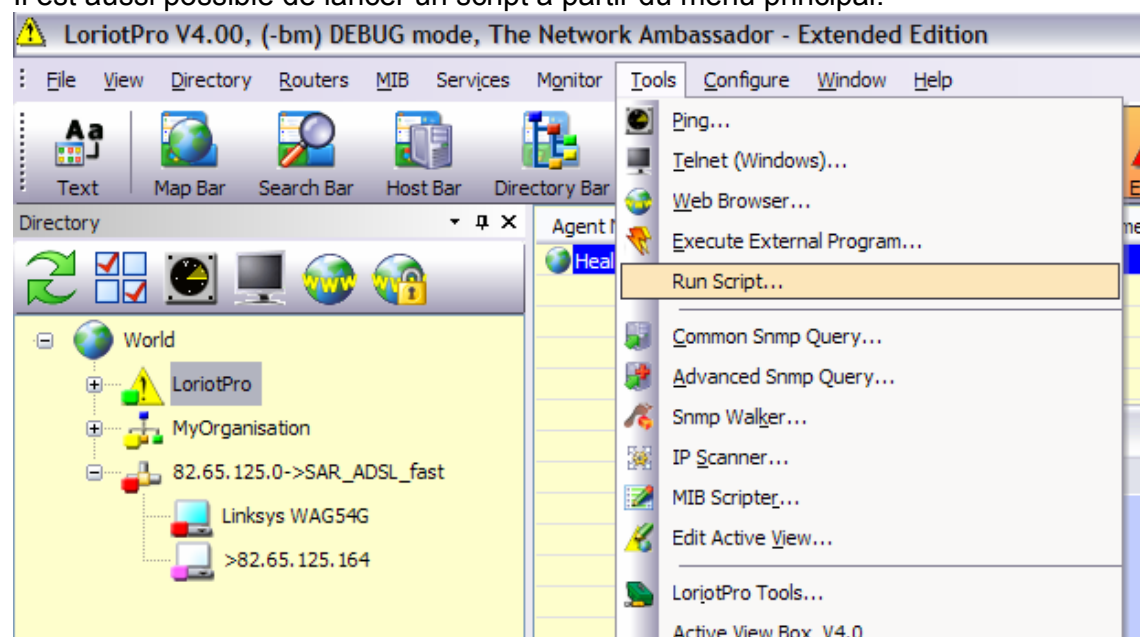
```

Value Name	Value in	Value out



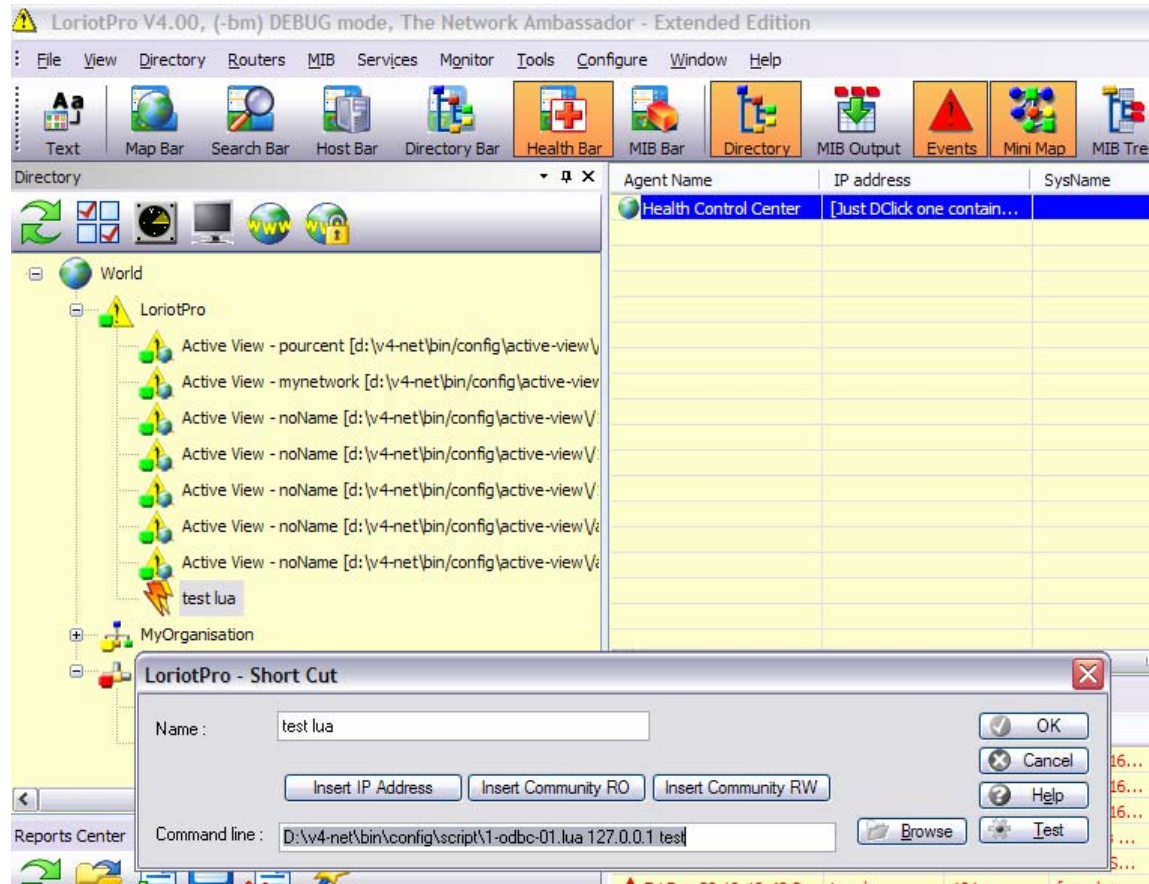
Menu principal

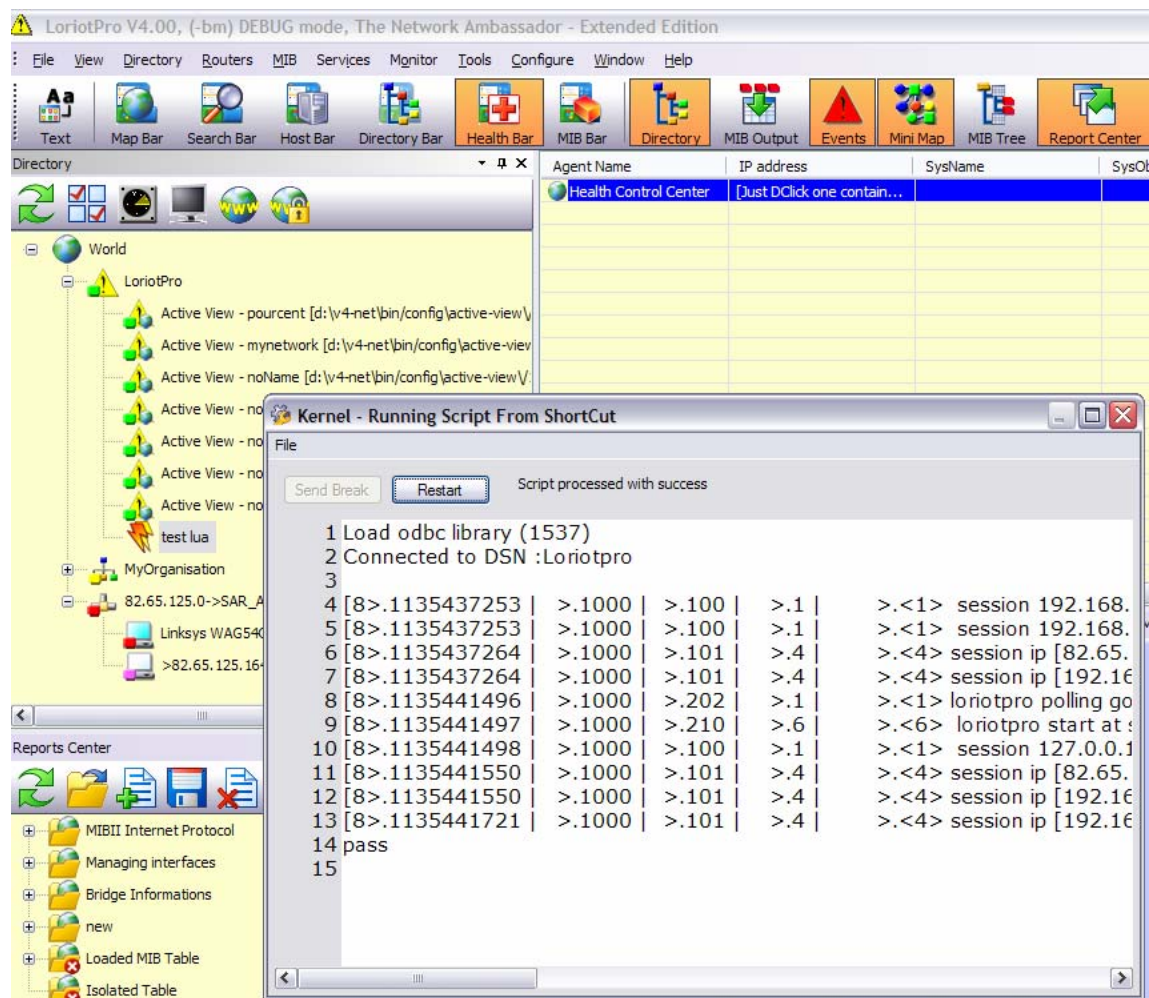
Il est aussi possible de lancer un script à partir du menu principal.



A partir d'un Shortcut de host

On peut définir un "shortcut" raccourci permettant de lancer un script associé à un host de la directory.





Attention

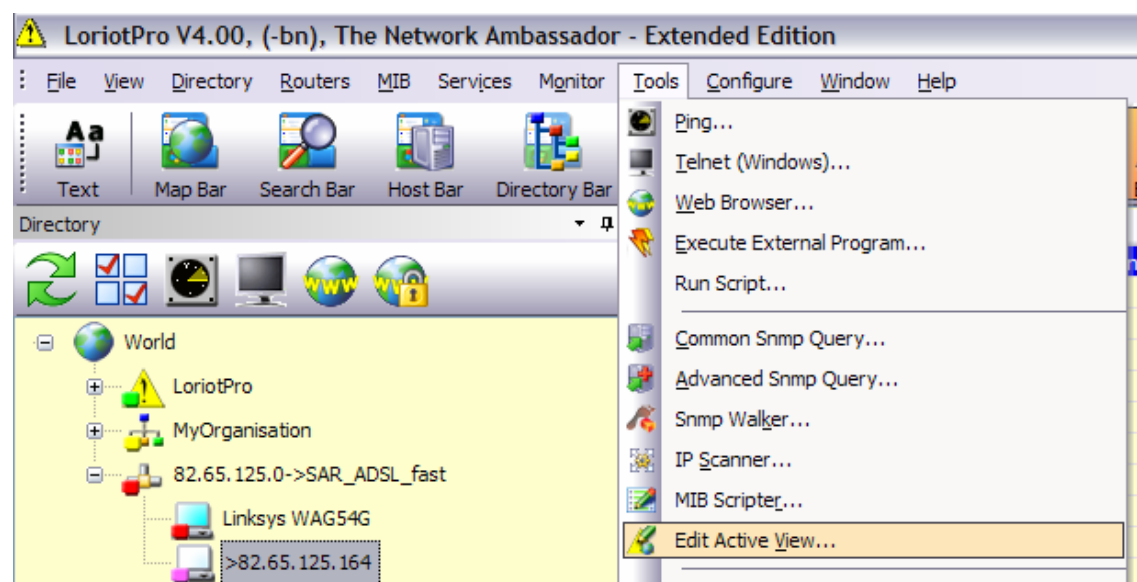
Pour des raisons de portabilités de la configuration d'un équipement de développement vers un équipement de production vous pouvez supprimer une partie du nom de fichier (le path) pour travailler avec des emplacements relatifs de fichier.

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\ correspondante au path du logiciel peut être supprimé.

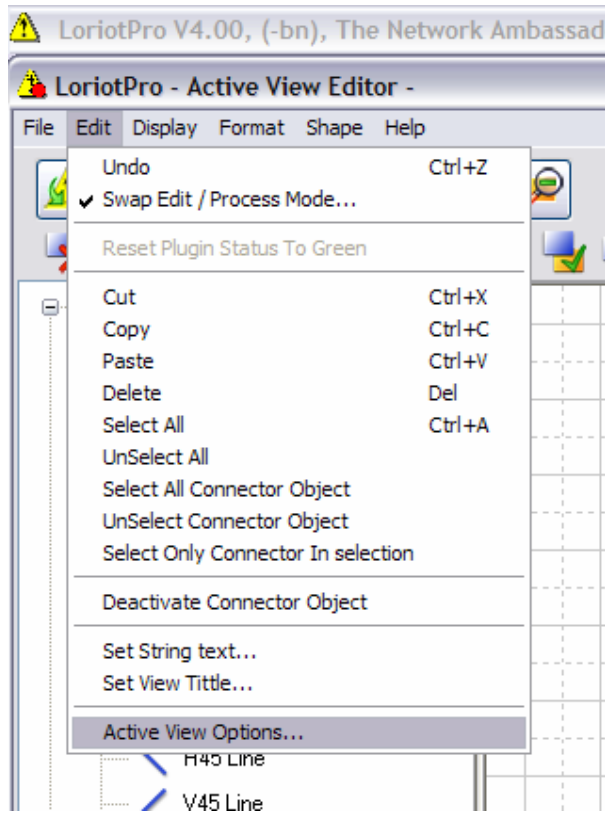
ShortCut associant une ActiveView Modal et un Script LUA

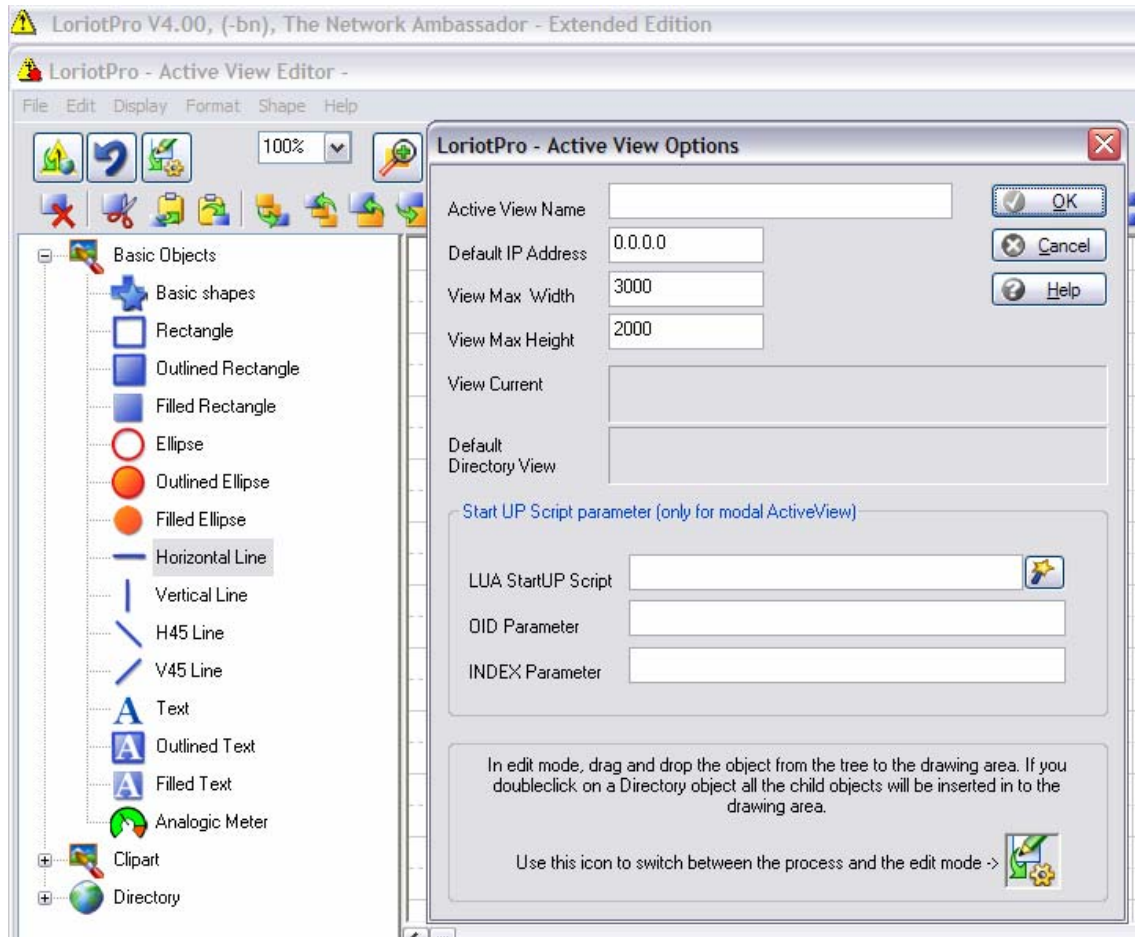
La fonction shortcut associé à un host de la directory permet aussi de lancer une ActiveView Modal qui pourra être configuré pour lancer au démarrage un script qui initialisera l'ActiveView. Dans l'exemple suivant une ActiveView est créée à partir de l'éditeur et précisant le chargement d'un script permettant la création automatique d'une carte « MAP » de niveau 3 (routeurs - réseaux).



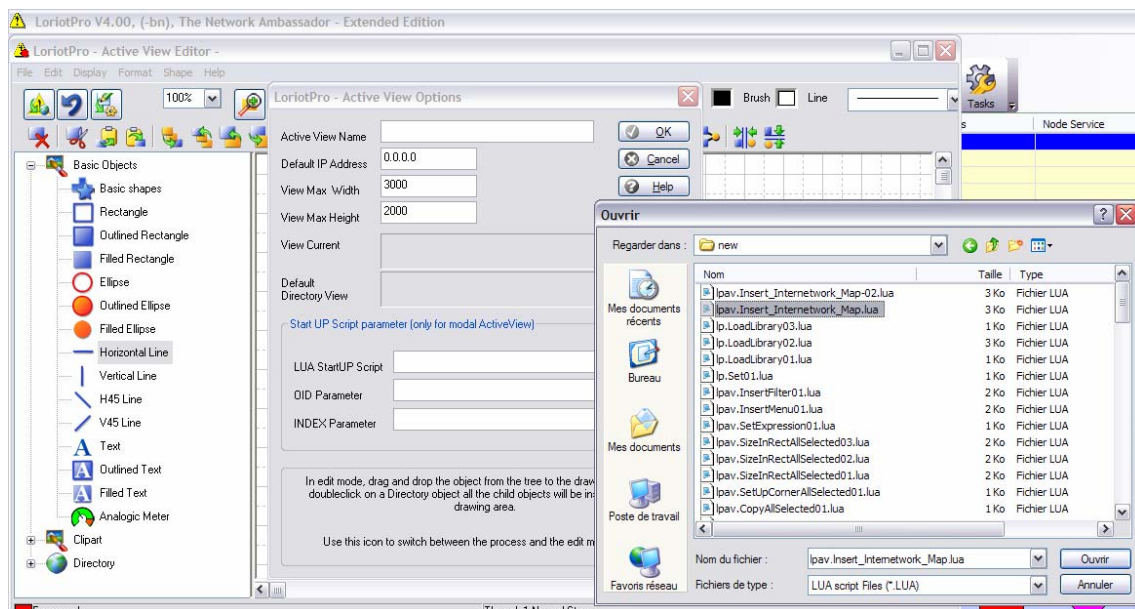
Une fois l'ActiveView ouverte choisissez l'option :

Edit>Active View Options

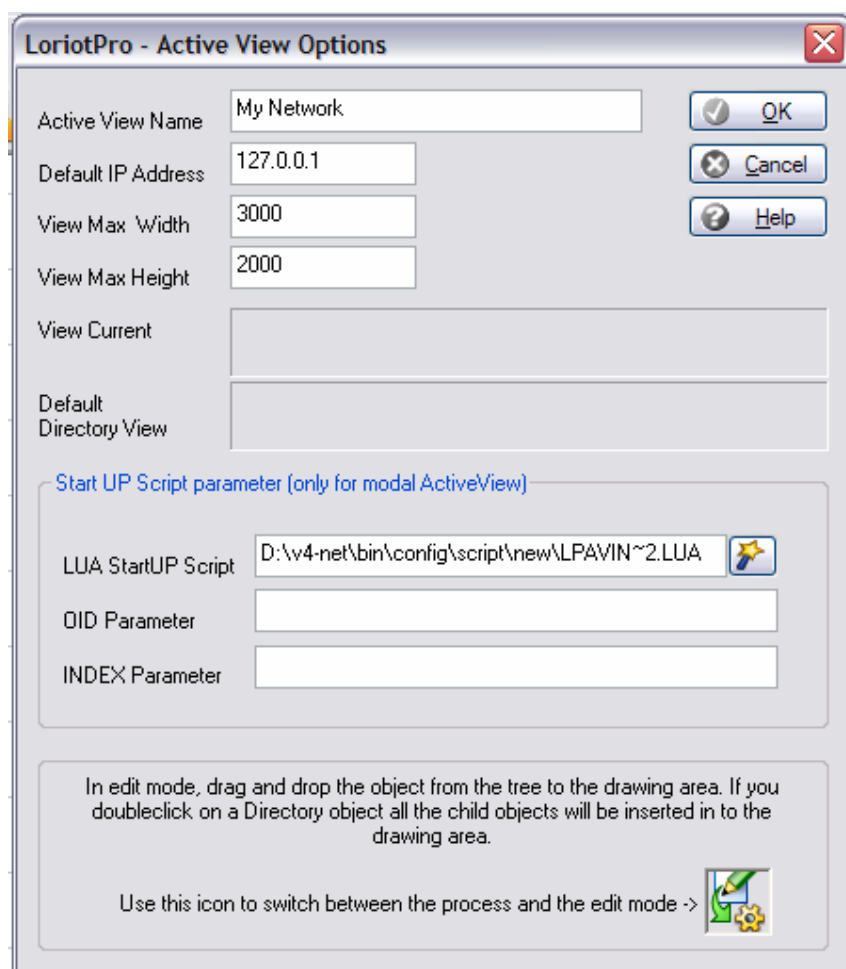




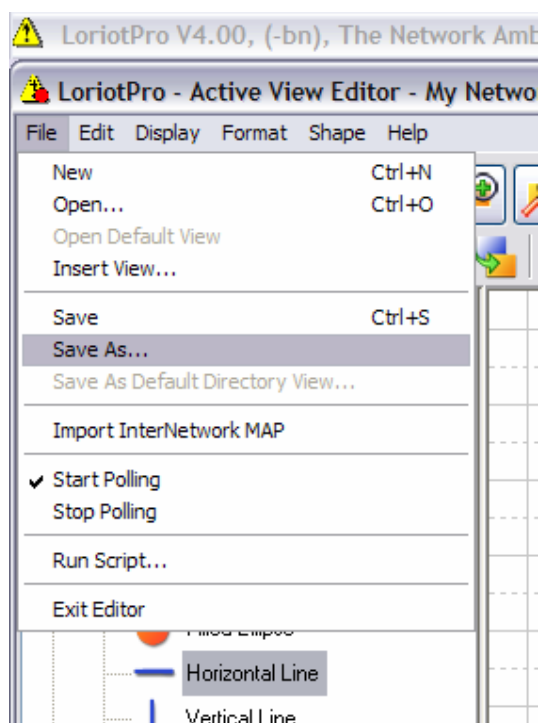
Les options LUA StartUP Script vous permettent de définir un script de démarrage pour votre future ActiveView :



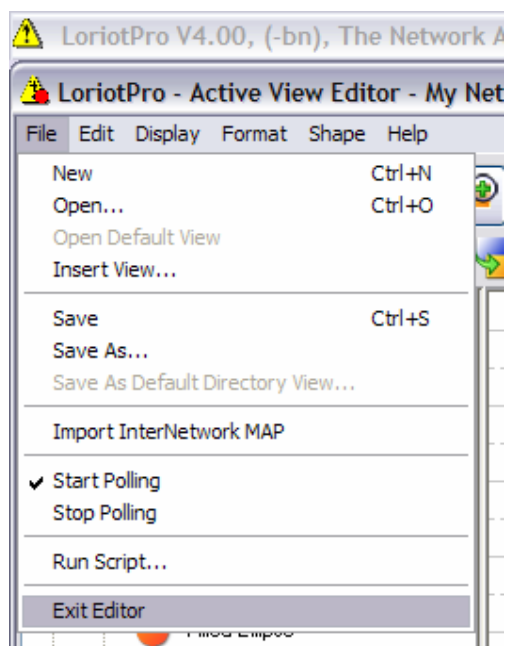
Nous choisissons le script **lpav.Insert_Internetnetwork_Map.lua**.



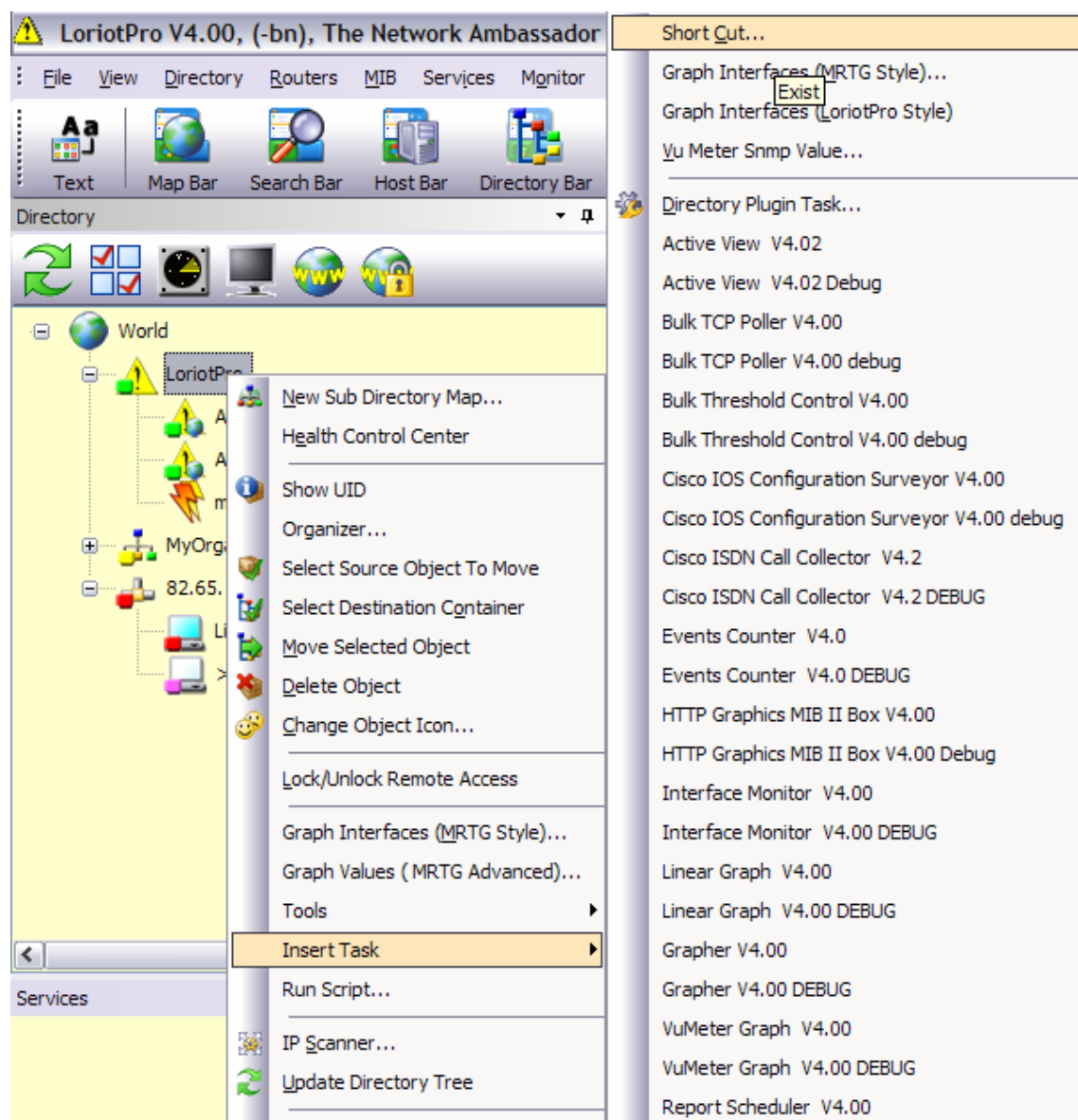
Le nom du fichier est transformé en mode court pour éviter les caractères blanc.



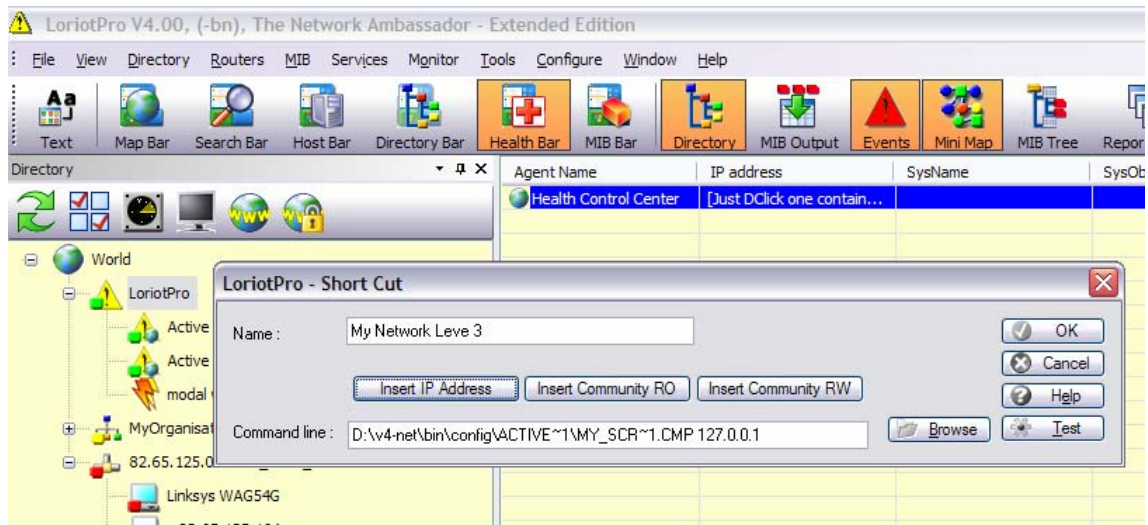
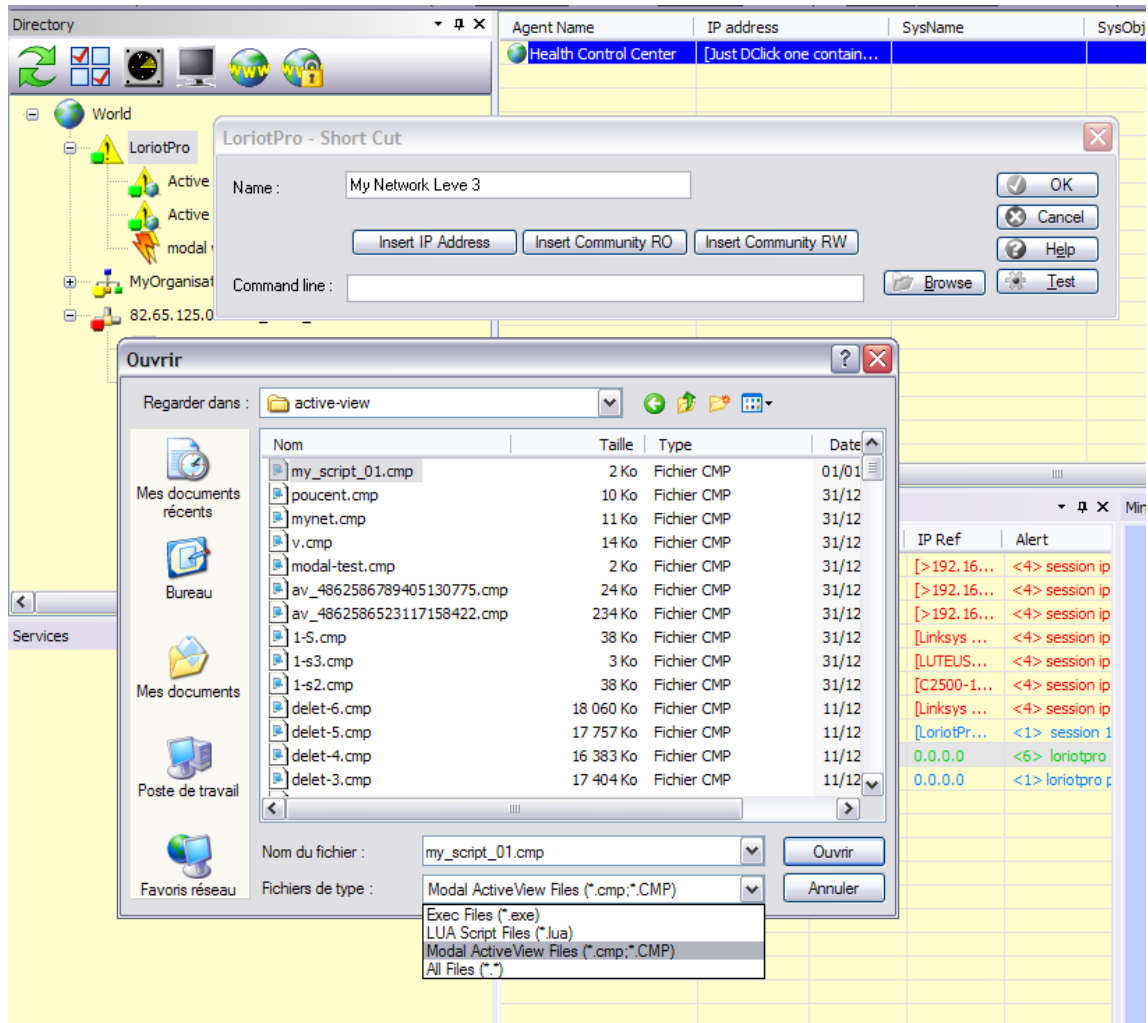
On sauvegarde l'ActiveView sous my_script_01.cmp.



On quitte l'éditeur.

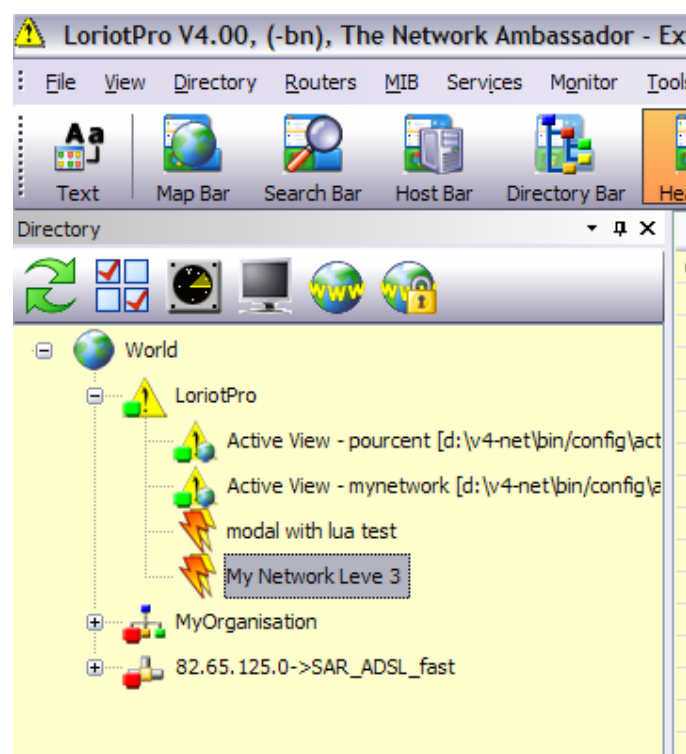


On insère un shortcut sous le host désiré, dans notre cas le 127.0.0.1.



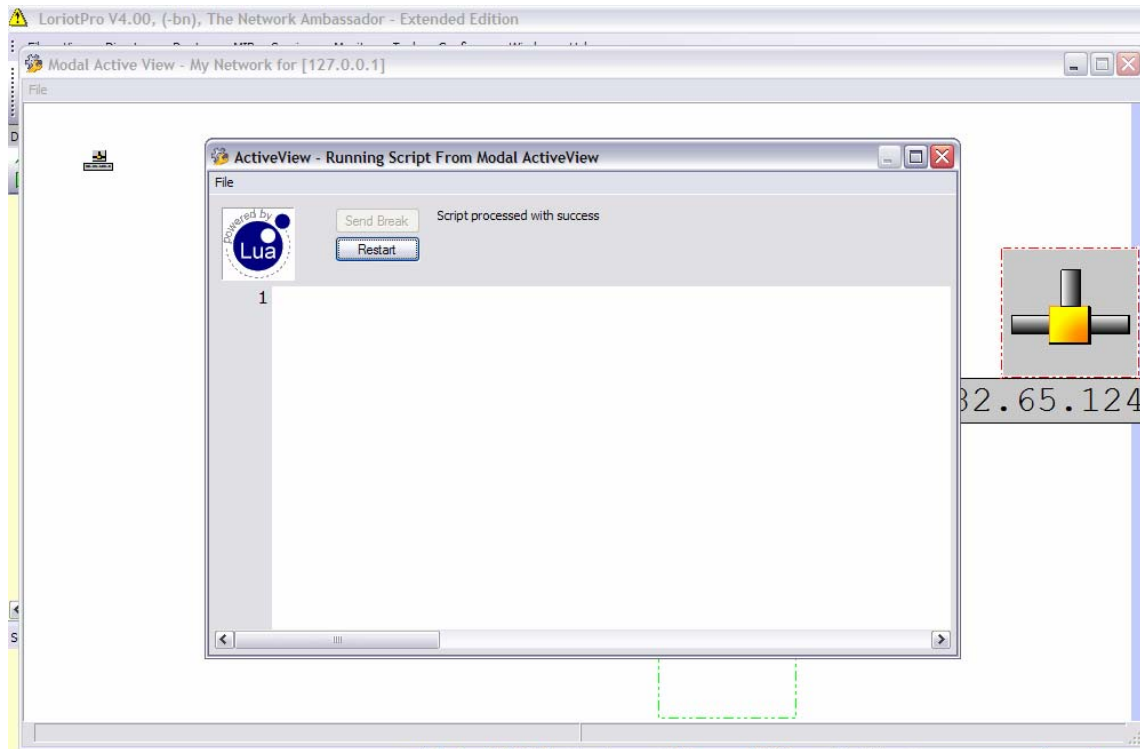
Attention

Le nom du fichier est mis en mode court et il faut impérativement mettre l'adresse IP désirée en deuxième paramètre.

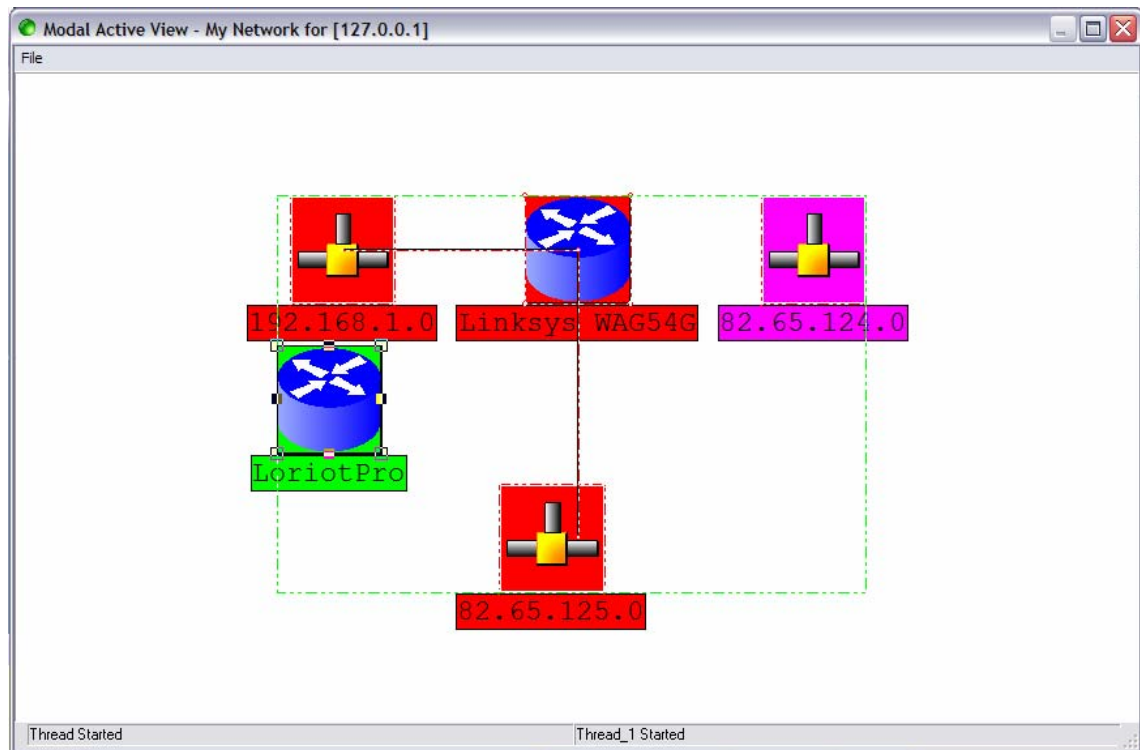


Dans notre exemple, un nouveau ShortCut est créé sous le host LoriotPro.

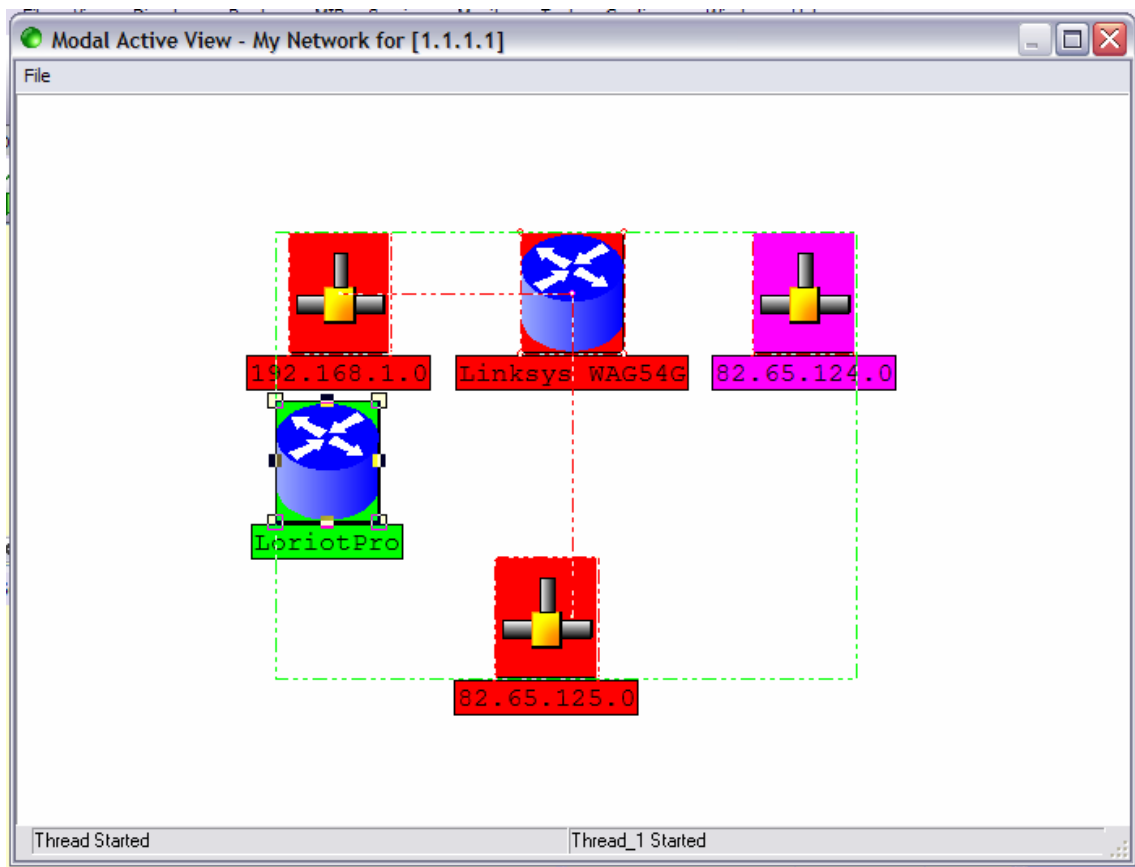
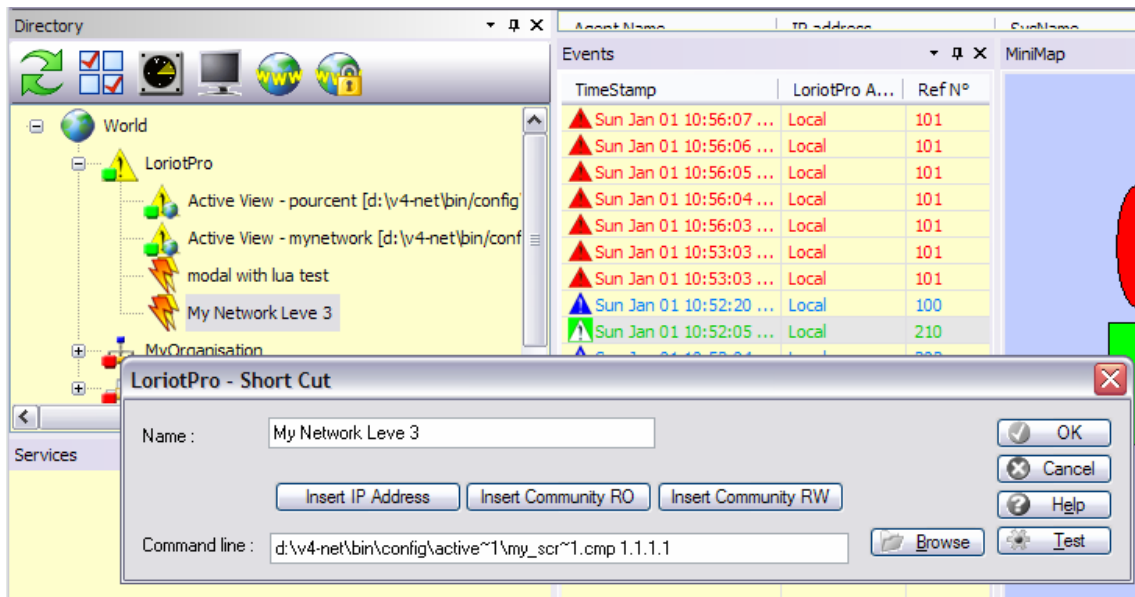
Si vous double cliquer sur l'Item « My Network Level 3 » L'ActiveView **my_script_01.cmp** est affichée et le script de démarrage **lpav.Insert_Internetwork_Map.lua** associé est lancé.



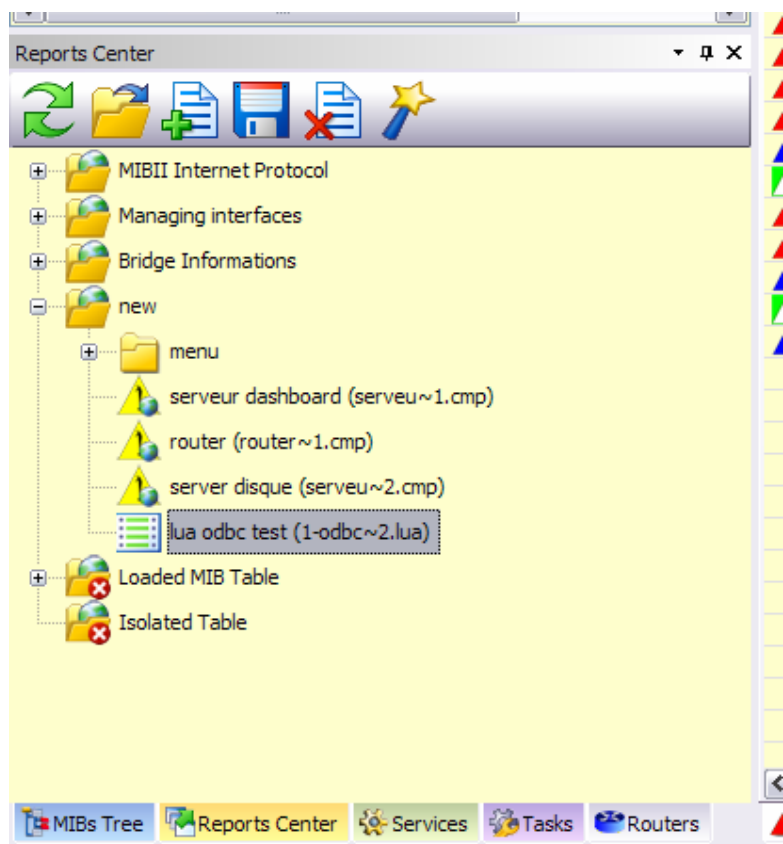
La fenêtre de script est automatiquement fermée à la fin du processus pour laisser la place à l'ActiveView.



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 de lancer une ActiveView pour l'adresse IP 1.1.1.1 sous le Host 127.0.0.1.



A partir du Report Center

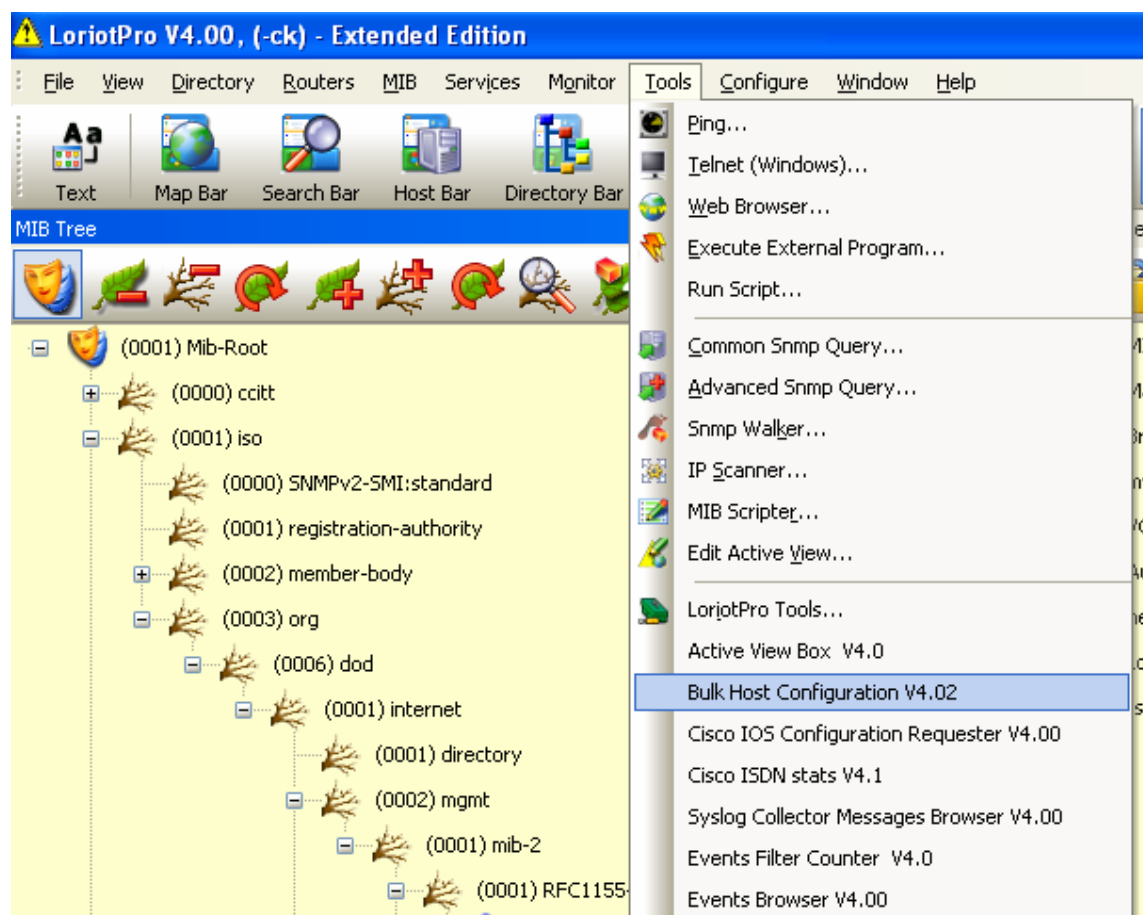


Les scripts mis en place dans le Report Center sont automatiquement associés au host par défaut sélectionné dans la Directory ou n'importe quel autre point du logiciel.

A partir du module de management des hosts (Bulk Configuration)

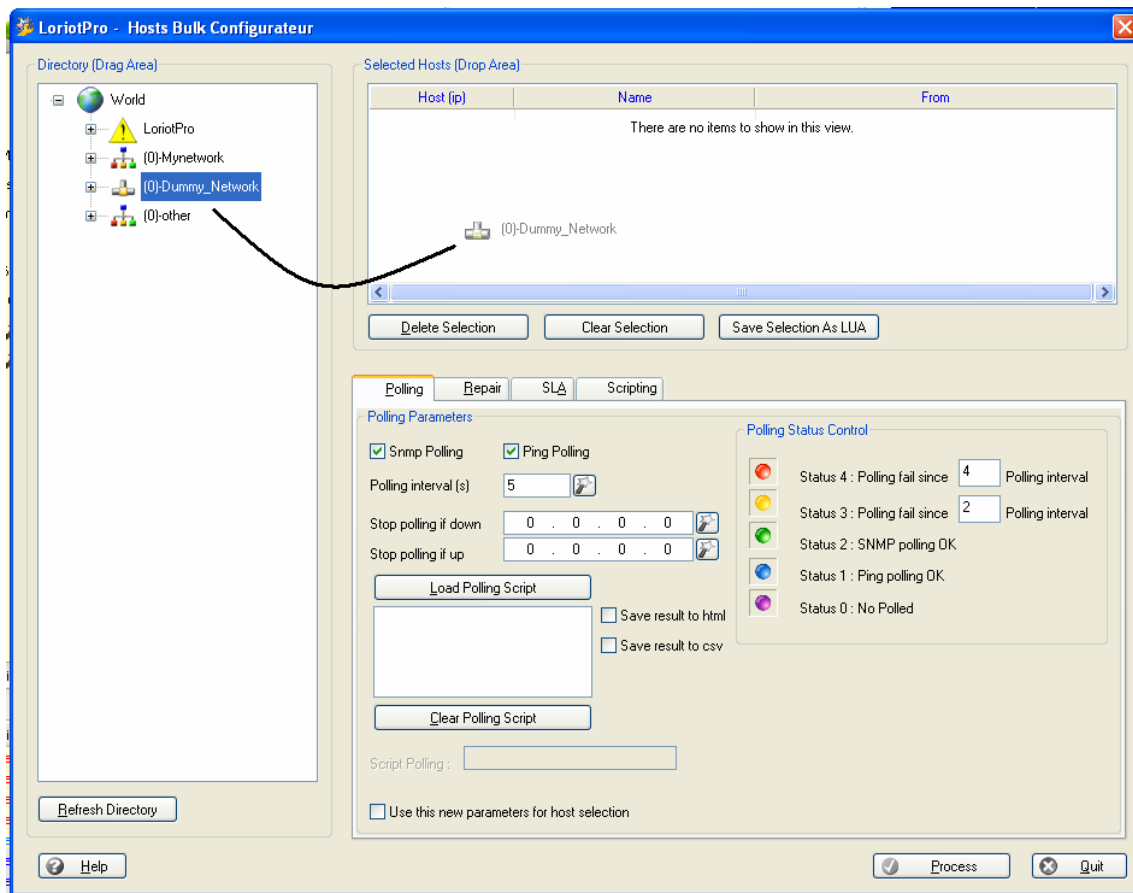
Le module de configuration par lot peut être utilisé pour lancer des scripts pour un ensemble de host présélectionnés.

Lancement du module :

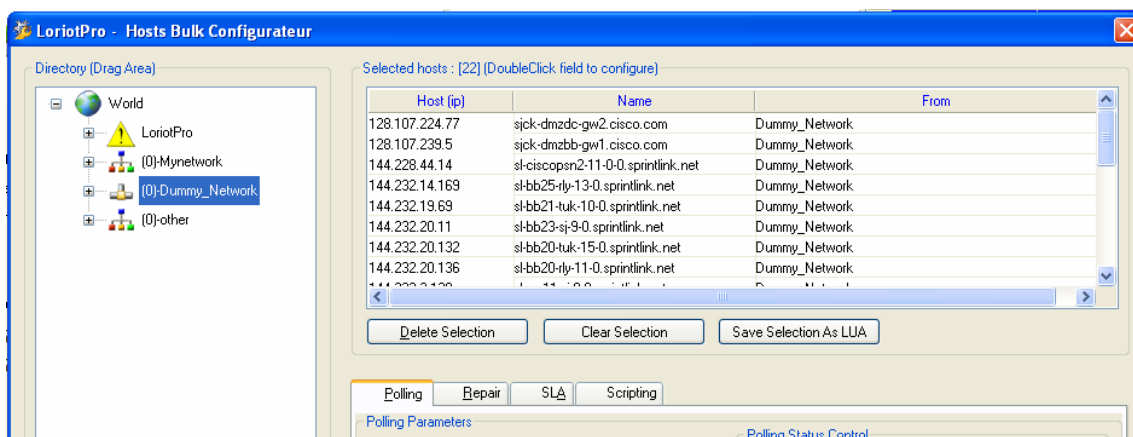


Ce module permet de réaliser des configurations ou des tâches de management des host par lot.

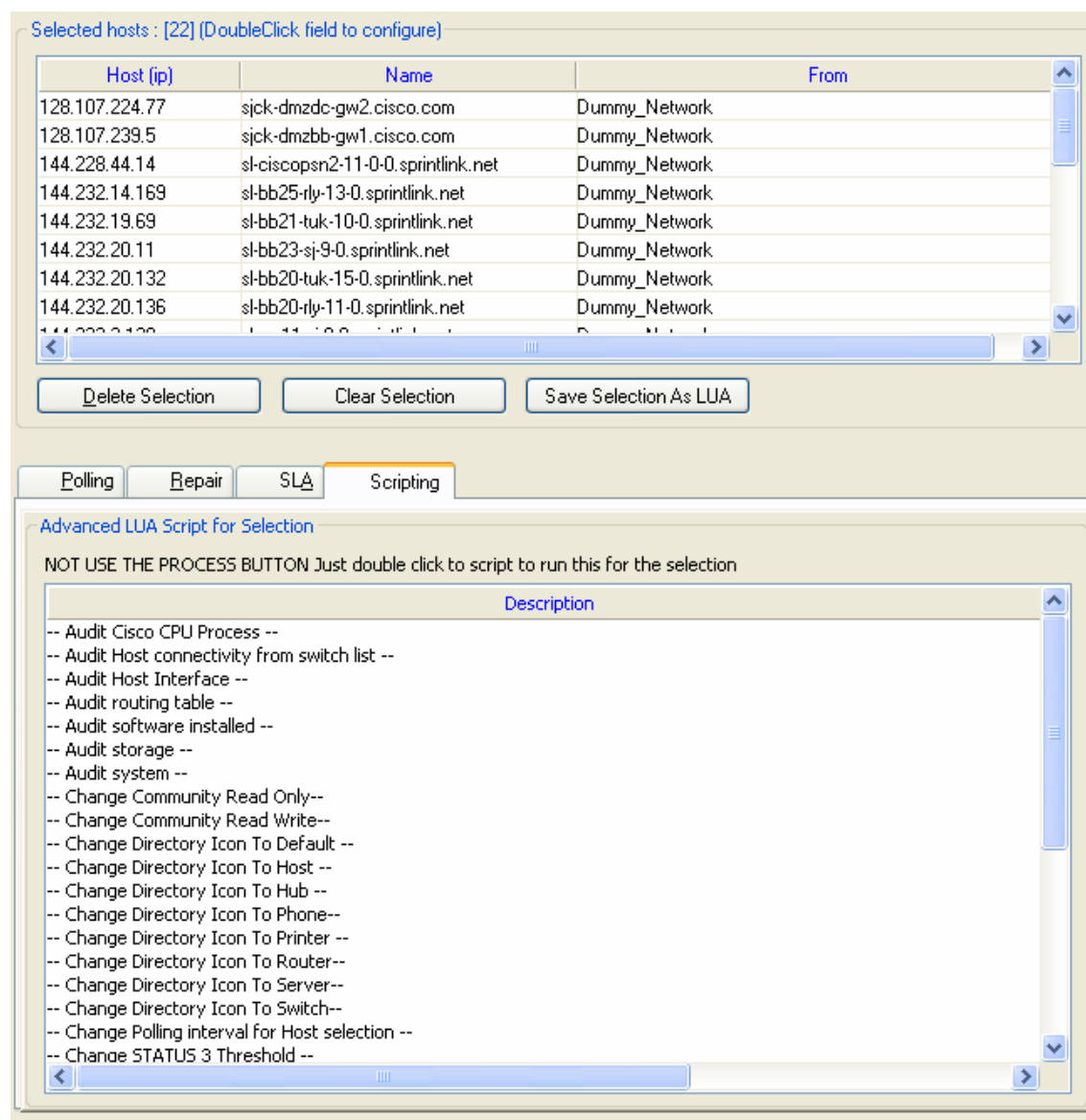
LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA



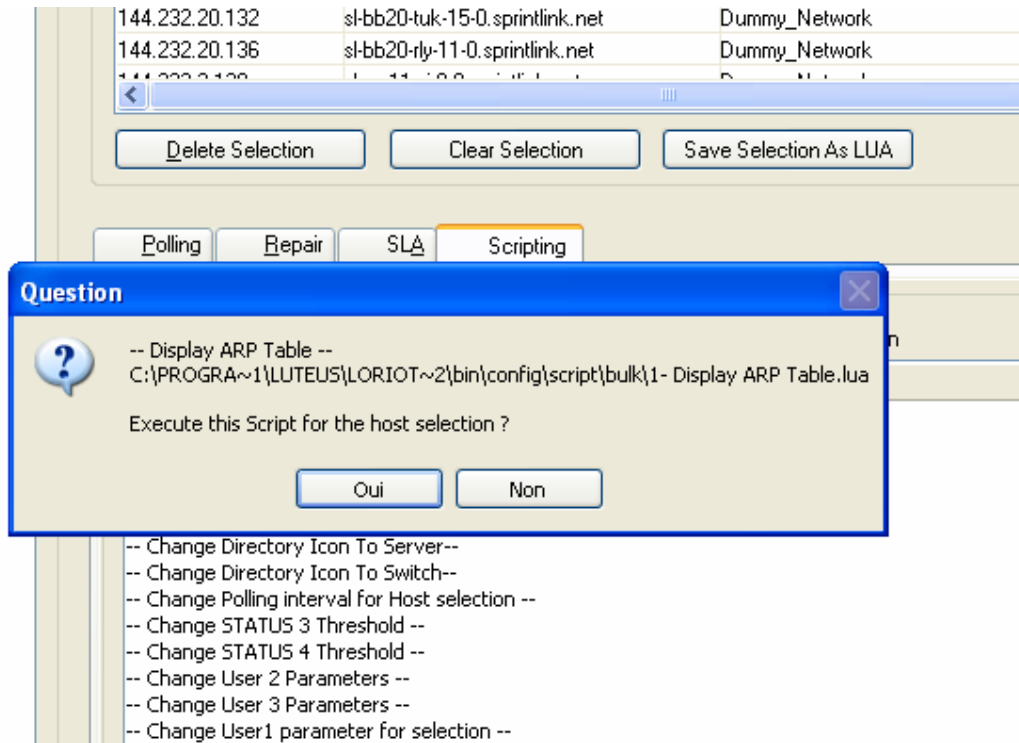
Il suffit de glisser les hosts de l'arbre vers la zone de sélection pour créer une liste de hosts.



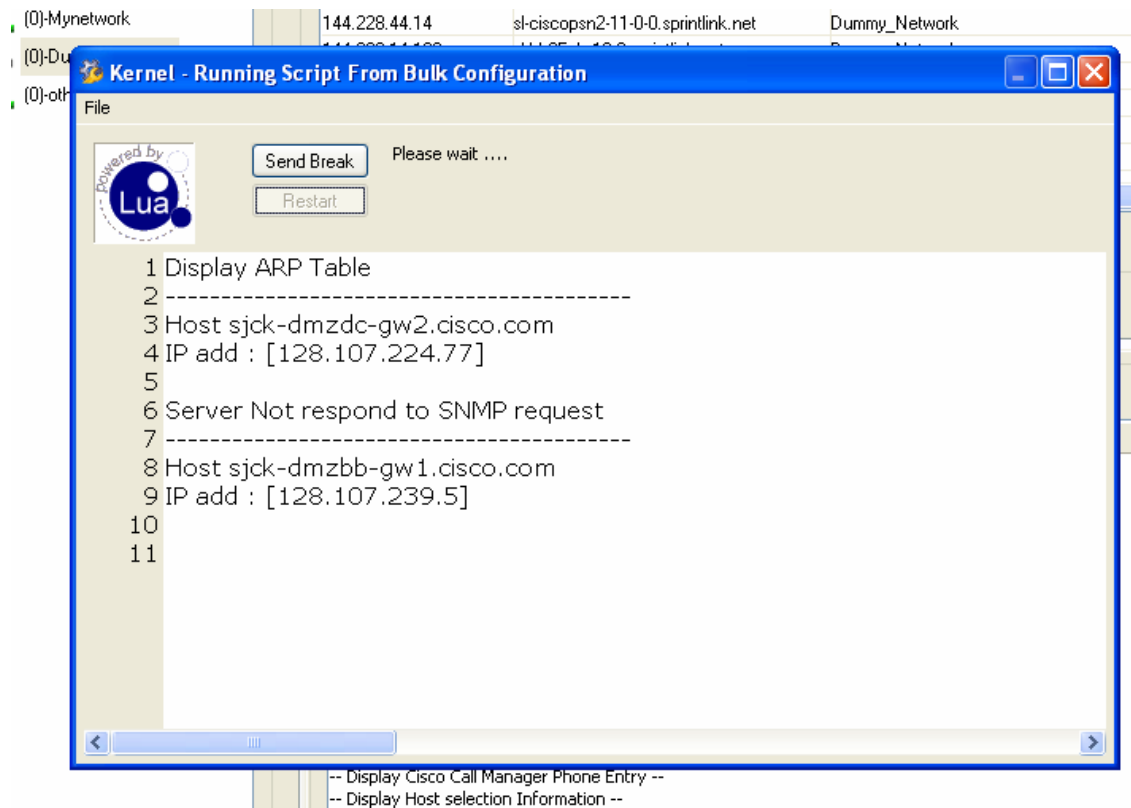
L'onglet Scripting permet d'avoir accès à une liste de script pré écrit permettant une gestion par lot.



Un simple double click sur une des lignes du listbox lance le script en l'associant à la liste de hosts.

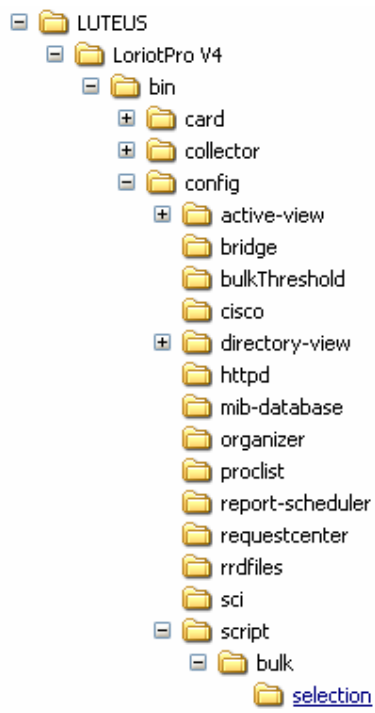


le module de lancement de script est appelé

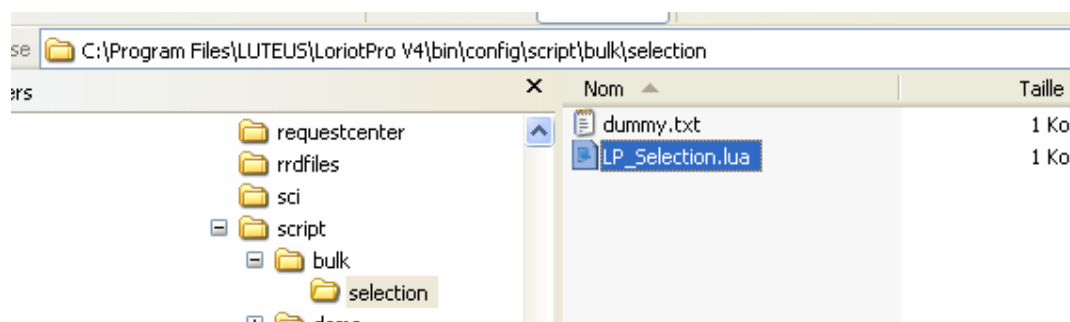


Création de nouveau script de type « bulk »

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)) prés configuré qui contient la liste des hosts sélectionnés.

```
-- Loriotpro V4
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 de host par lot.

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 utilisé pour définir l'intitulé du script dans le 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
```

```
Ip.Print("Host Number : ",hostnumber,"\n");

for j=0,hostnumber-1 do

    Ip_value=Ip.GetIPInformation(hosts[j],"aa");
    if Ip_value then
        Ip.Print("Modify : ",hosts[j]," ",aa.name,"\n");
        Ip.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
--]]

        Ip.SetIPOption(hosts[j],LP_LUA_ISDEVICE,5);
        Ip_value=Ip.GetIPInformation(hosts[j],"aaa");
        Ip.Print("\tNew type router : ",aaa["device_type"],"\n");

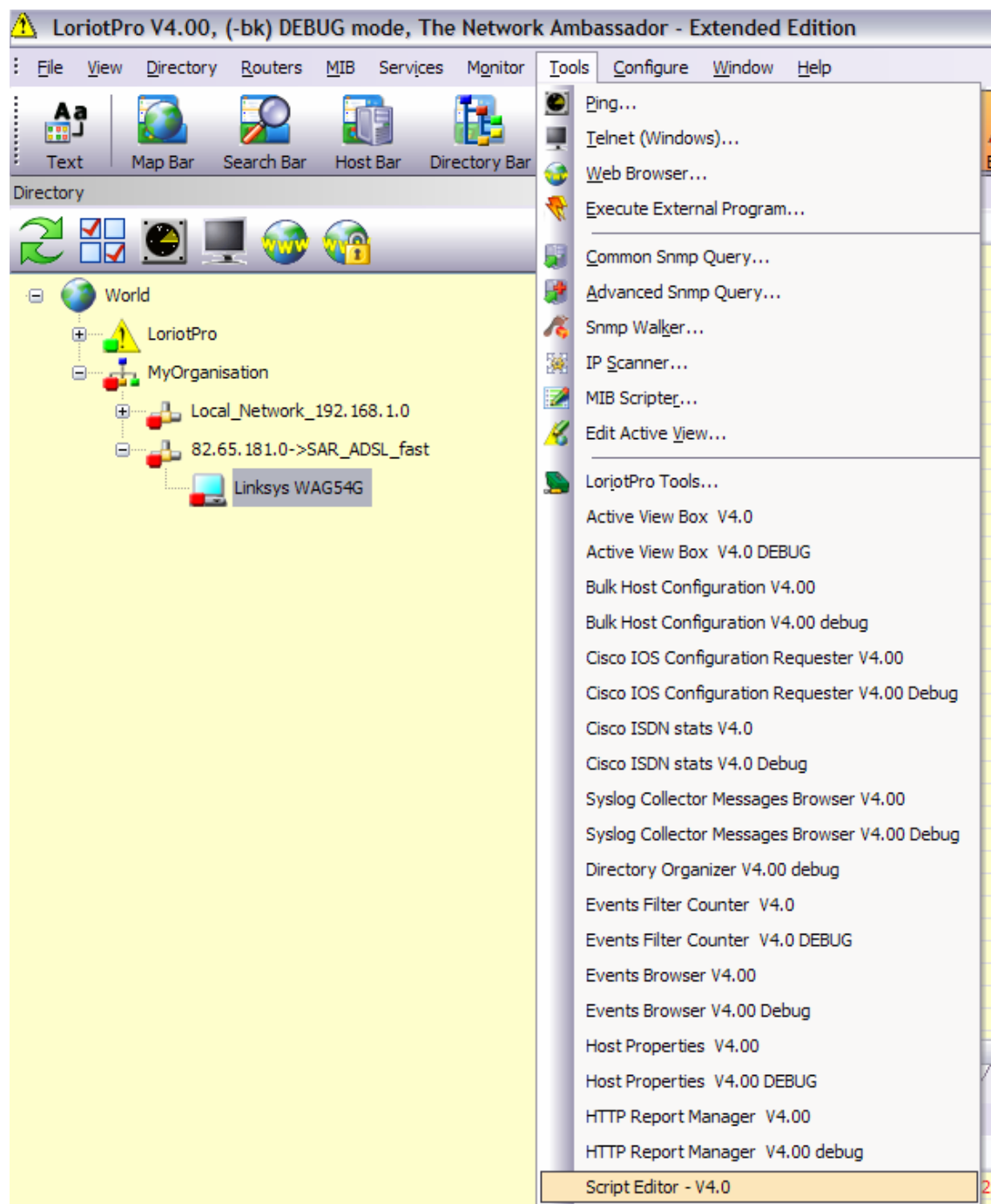
    else
        Ip_buffer="#ERROR";
        Ip.Print("no exist : ",hosts[j],"\n");
    end

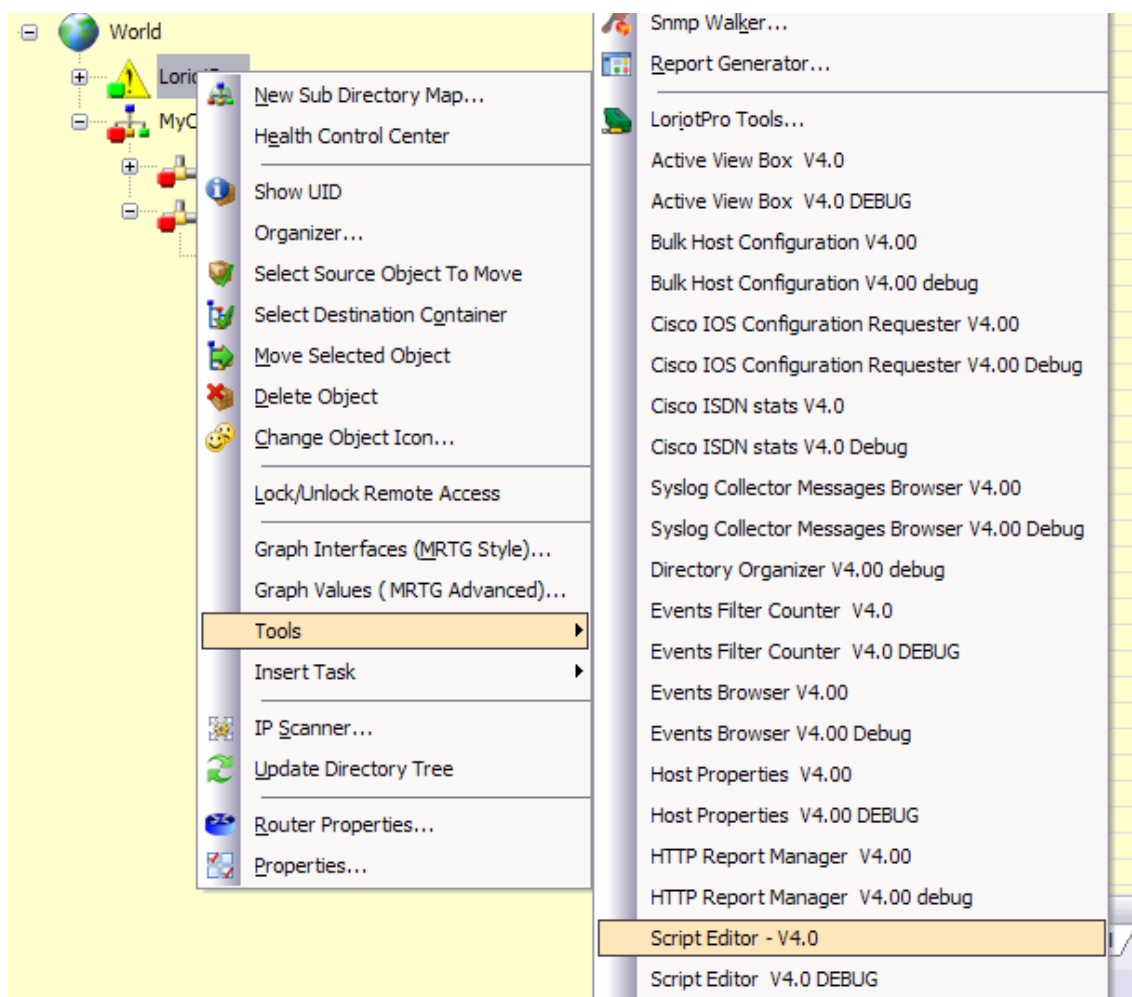
end
```

Plugin éditeur de script

Introduction

La version Etendue de LoriotPro inclut un nouveau plugin permettant d'éditer et de tester vos scripts. Ce plugin est appelé 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 qui vous seront bien utiles pour « déboguer » votre travail.





L'éditeur de script n'a pas la prétention d'être un grand éditeur, il est simplement là pour vous permettre de tester vos scripts dans l'environnement du logiciel.

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 quatre zones :

1. La zone d'édition du script
2. La zone d'édition des variables
3. La zone d'affichage des messages systèmes

4. La zone d'affichage des graphiques (ActiveView)

The screenshot displays the LoriotPro Script Editor interface. The main window shows a Lua script for inserting and configuring a router object. The script includes functions for refreshing the map, finding routers, and inserting objects with specific configurations.

```

34 lpav.RefreshMap();
35
36 router=lp.GetRouterFirst();
37 --while router~=nil do
38 --lp.Trace("router > "..net);
39 --if lp.GetRouterInformations(router,"a") ==1 then
40 --lp.Trace("router > "..a.name.."router.." ..a.x.." ..a.y);
41 lpav.InsertObject(LPAV_OBJ_ROUTER,a.name, a.x,a.y,a.UI
42 lpav.GetInformations("b1");
43 ip=lp.GetIPFirst();
44 while ip~=nil do
45 lp.GetIPInformation(ip,"aa");
46 if aa.groupref==lp.atopip(router) then
47 --lp.Trace("Found router");
48 net,mask=lp.GetNetworkFromIP(ip);
49 if net~=nil then
50 --lp.Trace("Found network");
51 lp.GetNetworkInformations(net
52 --lp.Trace("pass getnetworkinform
53 co=lpav.FoundFromAssoci
54 if co~=nil then
55 lpav.GetInformations("b2"
56 --lp.Trace("pass GetInformat
57 c3=lpav.InsertObject
58 --lp.Trace("Insert Connector > "..b2.UID.." ..b1.UID);
59 lpav.SetOption(LPAV_STATUS ,10);
60 lpav.SetC
61 lpav.SetC

```

The ActiveView Simulator Window shows a network diagram with a central router node labeled 'LoriotPro' and several peripheral nodes connected by lines. The diagram is enclosed in a green dashed box.

The Trace Window shows the following log output:

```

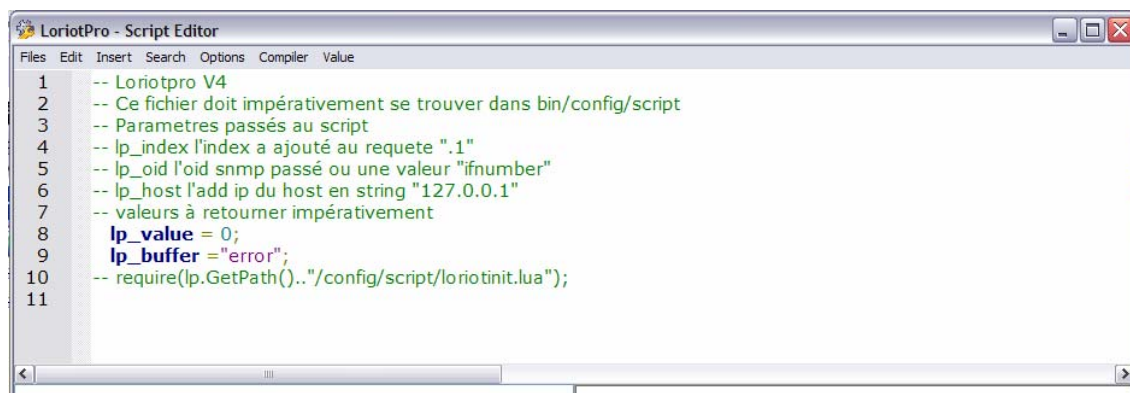
[192.168.1.0255.255.255.0]
[82.65.181.0255.255.255.0]

```

At the bottom left, a table displays variable values:

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		error
lp_value		0

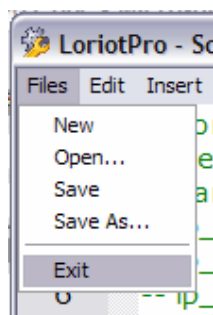
Zone d'édition du script



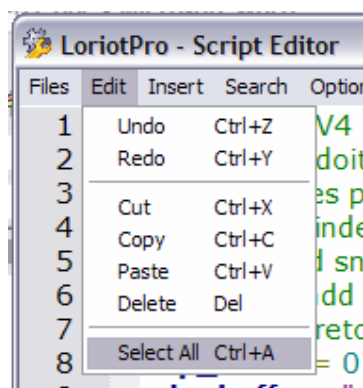
```
1 -- Loriotpro V4
2 -- Ce fichier doit impérativement se trouver dans bin/config/script
3 -- Parametres passés au script
4 -- lp_index l'index a ajouté au requete ".1"
5 -- lp_oid l'oid snmp passé ou une valeur "ifnumber"
6 -- lp_host l'add ip du host en string "127.0.0.1"
7 -- valeurs à retourner impérativement
8 lp_value = 0;
9 lp_buffer = "error";
10 -- require(lp.GetPath().."/config/script/loriotinit.lua");
11
```

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 « bleue gras ».

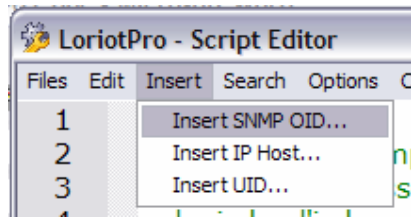
Vous pouvez importer/exporter vos scripts avec le menu Files :



Les différentes fonctions classiques de copier coller sont supportées :



Des modes Wizard permette d'insérer dans le texte des objets SNMP ou LoriotPro.



Ces objets seront insérés à l'emplacement du curseur.

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.

Zone d'édition des variables

Par défaut les scripts LUA LoriotPro utilisent 4 variables en entrée et 2 variables en sortie.

LoriotPro fournit les paramètres suivant au script :

Paramètres	Descriptions
Ip_host	L'adresse IP du host sous la forme d'une chaîne de caractères
Ip_oid	Le nom de l'objet snmp virtuel sous la forme d'une chaîne de caractères
Ip_index	L'index associé à l'objet snmp virtuel sous la forme d'une chaîne de caractères incluant le point. « .2 »
Ip_trace	1 active le mode trace 0 le stop. Lorsque le script tourne en production le mode trace est 0 par défaut.

Le script fournit les paramètres suivant à LoriotPro :

Paramètres	Descriptions
Ip_value	Une valeur (format double pour supporter le 64 bits) (si l'objet est de type « integer »)
Ip_buffer	Une chaîne de caractères (si l'objet est de type string)

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

Value Name	Value in	Value out
Ip_host	127.0.0.1	
Ip_index		
Ip_trace	1	
Ip_oid		
Ip_buffer		
Ip_value		

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 ».

The screenshot shows two parts of the application interface. The top part is a script editor with a menu open. The menu options are: "Go In Thread F5" (highlighted), "Stop Thread", "Kill Thread", and "Trace" (checked). The script editor shows four lines of code:

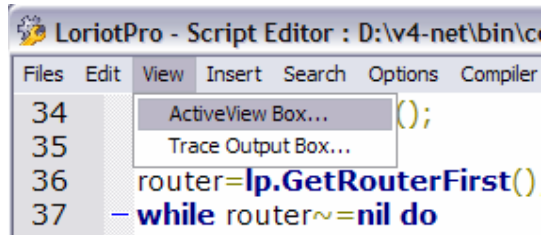
```
1 -- Loriotpro V4
2 -- Ce fichier doit in
3 -- Parametres pas
4 -- Ip_index l'index
```

The bottom part is a table with three columns: "Value Name", "Value in", and "Value out". The table contains the following data:

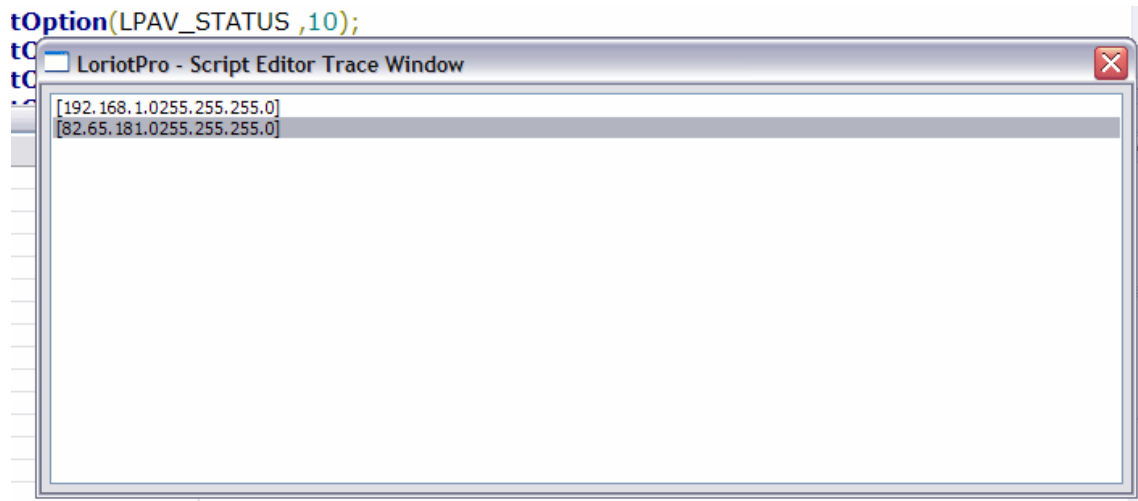
Value Name	Value in	Value out
Ip_host	127.0.0.1	127.0.0.1
Ip_index		
Ip_trace	1	1
Ip_oid		
Ip_buffer		good
Ip_value		0

Zone 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*** ») ;).



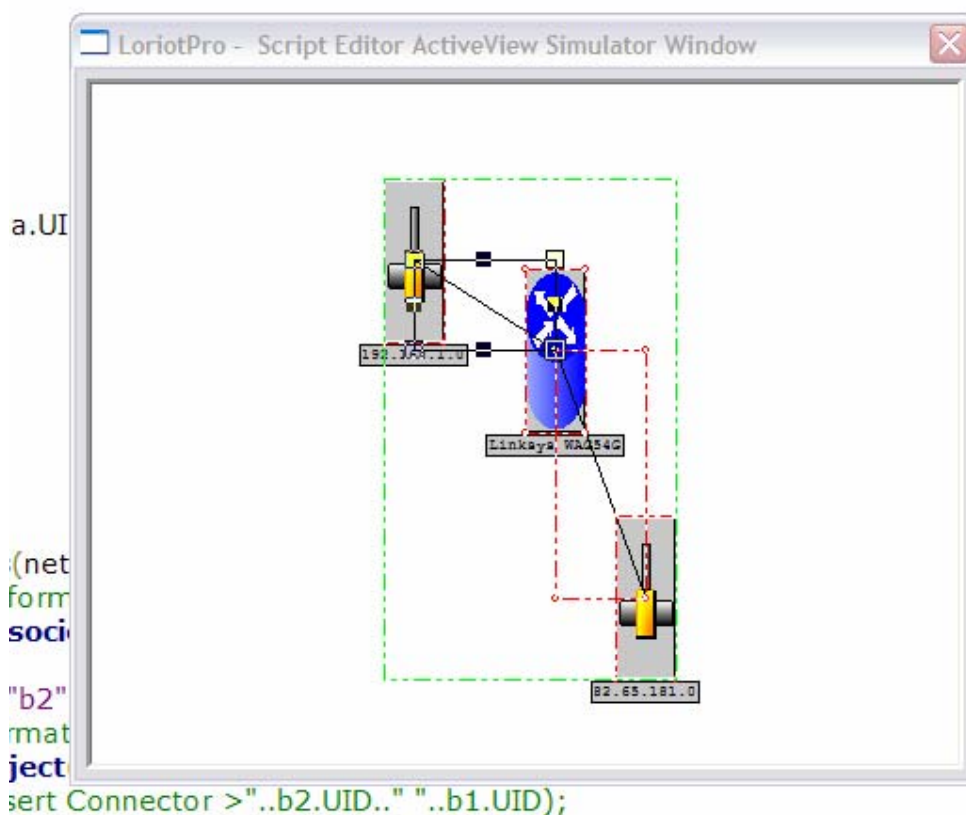
```
34  
35  
36 router=Ip.GetRouterFirst()  
37 while router~=nil do
```



Zone d'affichage des graphiques (ActiveView)

Cette zone permet de simuler une ActiveView, un grand nombre de fonction de la librairie **lpav** utilise comme variable des objets vectoriels d'ActiveView.

```
LoriotPro - Script Editor : D:\v4-net\bin\c
Files Edit View Insert Search Options Compiler
34 ActiveView Box... ();
35 Trace Output Box...
36 router=lp.GetRouterFirst()
37 while router~=nil do
```



Librairies LUA de LoriotPro

Introduction

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

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

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

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

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

Reportez vous à la documentation de LUA 5.02 mise en annexe pour plus d'informations concernant la syntaxe de LUA. La suite de cette documentation donnera des exemples d'utilisation des bibliothèques mais ne fournira pas d'informations sur l'écriture du langage LUA.

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 bibliothèques. 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/loriotinit.lua");
```

loriotinit.lua

```
--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 setipoption
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_GROUPOPREF=9;
LP_LUA_POLLING_SNMP=10;
LP_LUA_POLLING_PING=11;
LP_LUA_POLLING=12;
LP_LUA_ON=1;
LP_LUA_OFF=0;

--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;
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_HIDDEN_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.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_SCRIPT =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.InsertFilter
LPAV_FILTER_CONDITION_EQUAL =0;
LPAV_FILTER_CONDITION_SUP =1;
LPAV_FILTER_CONDITION_INF =2;
LPAV_FILTER_CONDITION_DIF =3;
LPAV_FILTER_CONDITION_SUP_EQUAL =4;
LPAV_FILTER_CONDITION_INF_EQUAL =5;
LPAV_FILTER_CONDITION_FIND =6;
LPAV_FILTER_CONDITION_BAD_EXPRESSION =7;
LPAV_FILTER_CONDITION_ANY =8;

function DisplayIPInformation (ip)

lp_value=lp_GetIPInformation(ip,"aa");
if lp_value then
lp_Trace("Name>".aa.name);
lp_Trace("Session>////////////////////////////////////");
lp_Trace("IP>".aa["ip"]);
lp_Trace("ip_add>".aa["ip_add"]);
```

```

lp_Trace("name>".aa["name"]);
lp_Trace("status>".aa["status"]);
lp_Trace("communityro>".aa["communityro"]);
lp_Trace("communityrw>".aa["communityrw"]);
lp_Trace("user1>".aa["user1"]);
lp_Trace(aa["user2"]);
lp_Trace(aa["user3"]);
lp_Trace(aa["sysid"]);
lp_Trace(aa["sysname"]);
lp_Trace(aa["groupref"]);
lp_Trace(aa["isinrepair"]);
lp_Trace(aa["trapgenerated"]);
lp_Trace(aa["sla"]);
lp_Trace(aa["ri_ifgraph"]);
lp_Trace(aa["ri_ifinotets_p"]);
lp_Trace(aa["ri_ifoutoctets_p"]);
lp_Trace(aa["snmppooling"]);
lp_Trace(aa["pingpooling"]);
lp_Trace(aa["timeslast"]);
lp_Trace(aa["UID"]);
end
end

function DisplayUID( uid)
    lp_value=lp_GetUIDInformation(uid,"aaa");
    if lp_value then
        lp_Trace("UID>".uid.."////////////////////////");
        lp_Trace("name>".uid,aaa.name);
        lp_Trace("status>".uid,aaa.status);
        lp_Trace("plugin_name>".uid,aaa.plugin_name);
        lp_Trace("object_type".uid,aaa.object_type);
        lp_buffer =aaa.name;
    else
        lp_buffer ="#ERROR";
    end
end
end

```


LoriotPro (Ip) Librairie

Cette librairie est attachée à la dll **dsnmp_kernel42.dll**.

Les fonctions de la librairie ont une syntaxe du type :

`Ip.GetNext(« value. »,...)` ;

Tableau de définition de la librairie

```
static const luaL_reg dsnmplib[] = {
  {"Get", Ip_Get},
  {"GetNext", Ip_GetNext},
  {"Gets", Ip_Gets},
  {"GetRows", Ip_GetRows},
  {"GetPath", Ip_GetPath},
  {"Trace", Ip_Trace},
  {"Break", Ip_Break},
  {"GetIPFirst", Ip_GetIPFirst},
  {"GetNextIP", Ip_GetNextIP},
  {"GetIPInformation", Ip_GetIPInformation},
  {"SetIPOption", Ip_SetIPOption},
  {"GetUIDInformation", Ip_GetUIDInformation},
  {"GetChildUID", Ip_GetChildUID},
  {"atoip", Ip_atoip},
  {"iptoa", Ip_iptoa},
  {"GetIPFromMAC", Ip_GetIPFromMAC},
  {"FindIPFronDynamicDNS", Ip_FindIPFronDynamicDNS},
  {"SendEvent", Ip_SendEvent},
  {"SendExternEvent", Ip_SendExternEvent},
  {"IsLoadedMIBRef", Ip_IsLoadedMIBRef},
  {"FindSNMPObject", Ip_FoundSNMPObject},
  {"GetFirstNetwork", Ip_GetFirstNetwork},
  {"GetNextNetwork", Ip_GetNextNetwork},
  {"GetNetworkFromIP", Ip_GetNetworkFromIP},
  {"GetFirstRouter", Ip_GetFirstRouter},
  {"GetNextRouter", Ip_GetNextRouter},
  {"GetNetworkInformation", Ip_GetNetworkInformation},
  {"GetRouterInformation", Ip_GetRouterInformation},

  {NULL, NULL}
};
```

Ip.Get

```
value,buffer=Ip.Get("ip","oid");
```

Explication

Ip.Get réalise une interrogation SNMP de type GET d'un seul objet « oid » vers l'adresse « ip ». La fonction retourne deux valeurs :

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x. Attention cette adresse doit impérativement exister dans la Directory.

« oid »

C'est une chaîne de caractères contenant un unique OID SNMP. Attention cet OID doit impérativement exister dans votre database.

Paramètres en sortie

value :

Le résultat numérique si l'objet SNMP « oid » est de type integer/gauge... sinon 0 si il est de type octetstring.

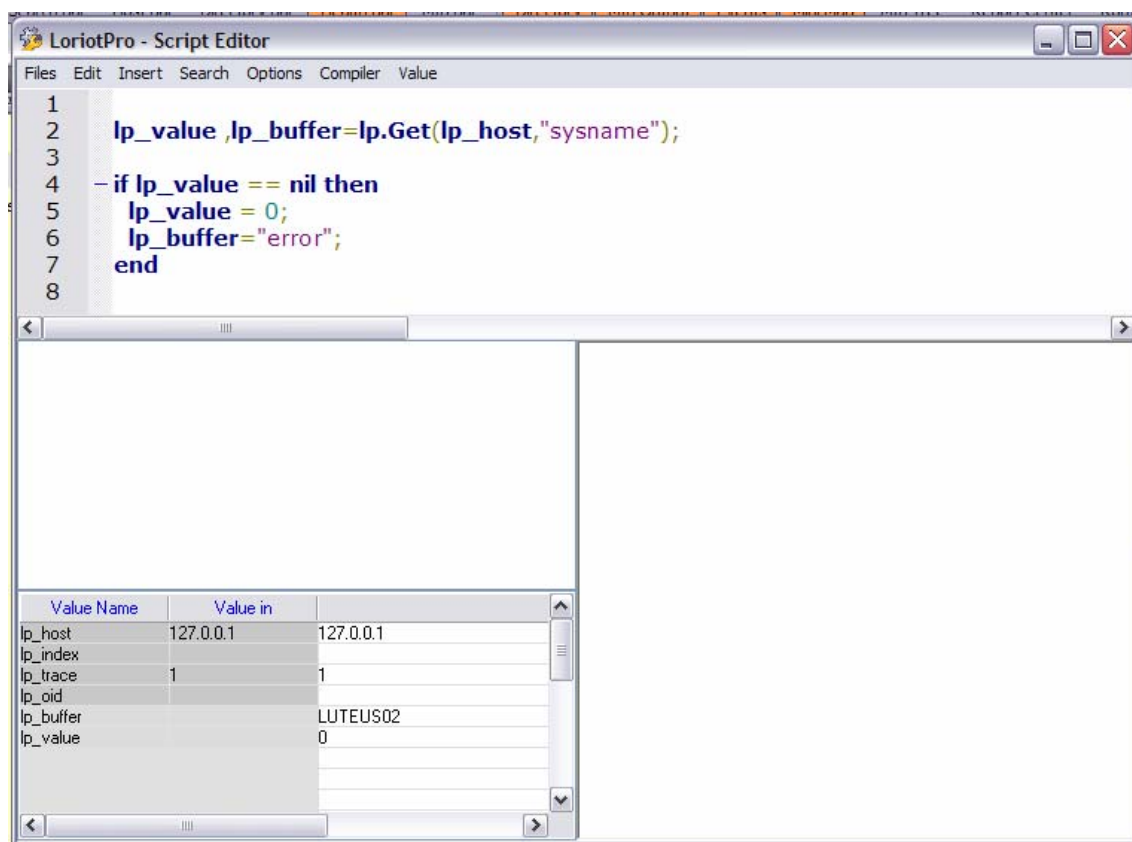
Renvoie **nil** si une erreur est rencontrée

buffer :

Contient sous la forme d'une chaîne de caractères le résultat du GET SNMP.

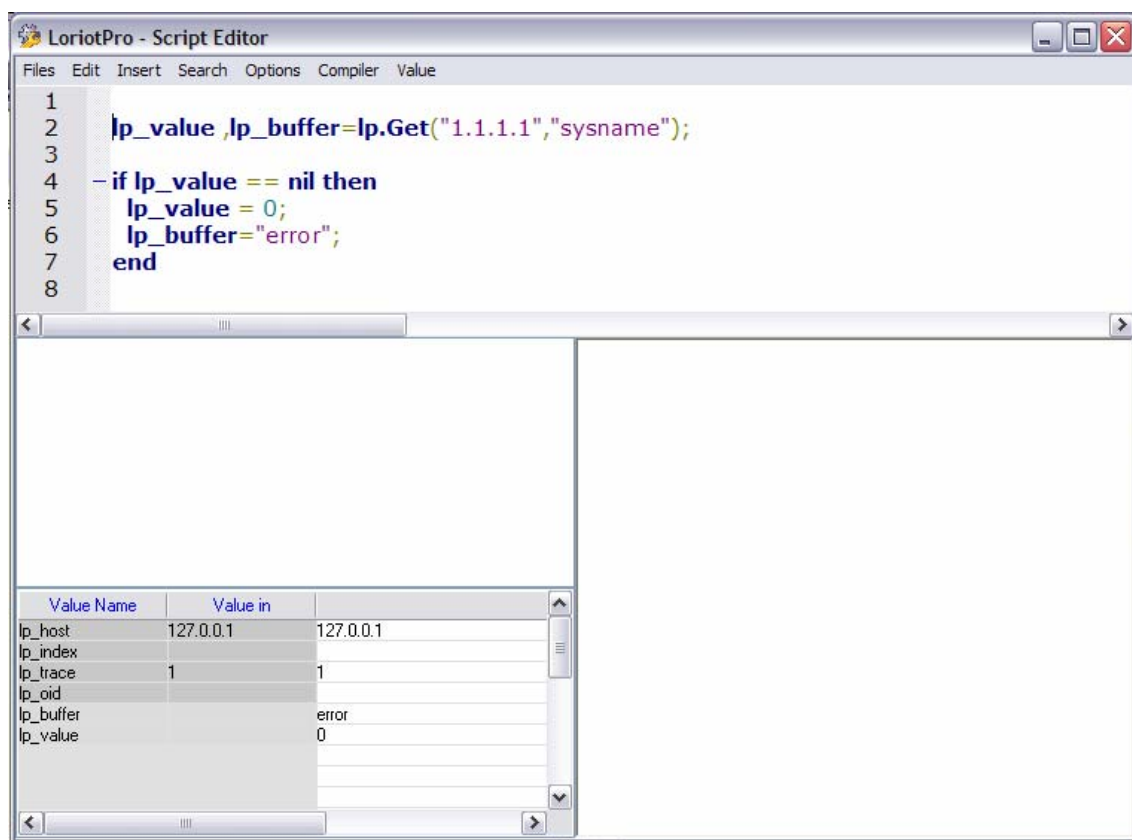
Renvoie **nil** si une erreur est rencontrée

Exemple :



Dans cet exemple le script utilise la variable **lp_host** passé par LoriotPro au script et demande l'OID SNMP *sysname*.

Les résultats sont affichés dans la colonne « Value out ».



Dans cet exemple le host d'adresse ip « 1.1.1.1 » n'existe pas dans la directory et la commande **lp.Get** retourne lp_value = nil .

Ip.GetBRC

```
value,buffer=Ip.GetBRC("ip","oid",vlanid);
```

Explication

Ip.GetBRC réalise une interrogation SNMP de type GET d'un seul objet « oid » vers l'adresse « ip ». La fonction retourne deux valeurs :

Cette fonction permet de collecter des informations dans les instances de MIB par Vlan de Cisco.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x. Attention cette adresse doit impérativement exister dans la Directory.

« oid »

C'est une chaîne de caractères contenant un unique OID SNMP.

Attention

cet OID doit impérativement exister dans votre database.

Vlanid le numéro de VLAN à qui s'adresse votre requête

Paramètres en sortie

value :

Le résultat numérique si l'objet SNMP « oid » est de type integer/gauge... sinon 0 si il est de type octet string.

Renvoi **nil** si une erreur est rencontrée

buffer :

Contient sous la forme d'une chaîne de caractères le résultat du GET SNMP.

Renvoi **nil** si une erreur est rencontrée

Ip.Set

```
value,buffer=Ip.Set("ip","oid");
```

Explication

Ip.Set réalise une modification de l'objet SNMP unique « oid ». La fonction retourne deux valeurs ou nul si une erreur est trouvée.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x.

Attention

Cette adresse doit impérativement exister dans la Directory et la community SNMP RW de « ip » doit être configuré et valide.

« oid »

C'est une chaîne de caractères contenant un unique OID SNMP et un codage permettant de fournir la nouvelle valeur.

Oidname.index/(type)value

syntax	Value type
(nu)	NUL
(id)	Object ID
(hs)	Hexa string
(os)	Octet string
(in)	Integer
(ip)	IP adresse
(co)	Counter
(ga)	Gauge
(ti)	Timeticks

Attention

OIDname doit impérativement exister dans votre MIB database.

Paramètres en sortie

value :

Le résultat numérique si l'objet SNMP « oid » est de type integer/gauge... sinon 0 si il est de type octetstring.

Buffer :

Le résultat de la commande Set sous la forme d'une chaîne de caractères.

Renvoie **nil** si une erreur est rencontrée

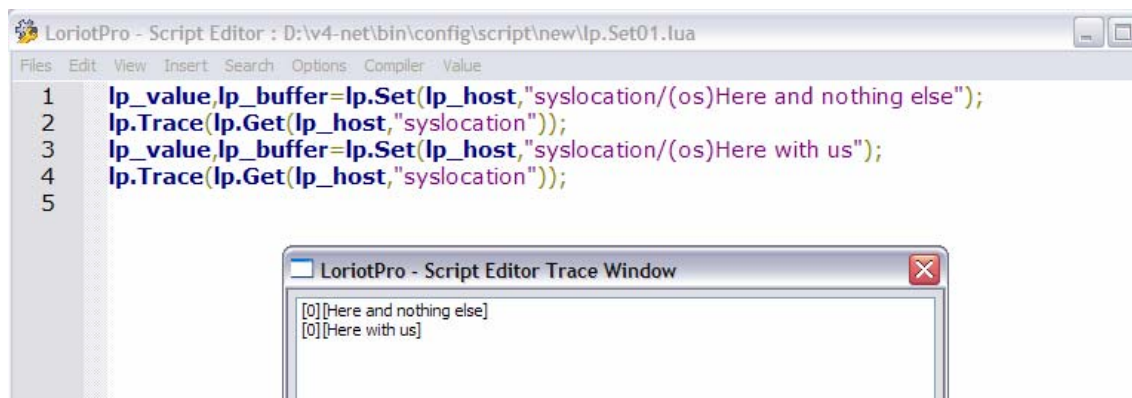
buffer :

Contient sous la forme d'une chaîne de caractères le résultat du GET SNMP.

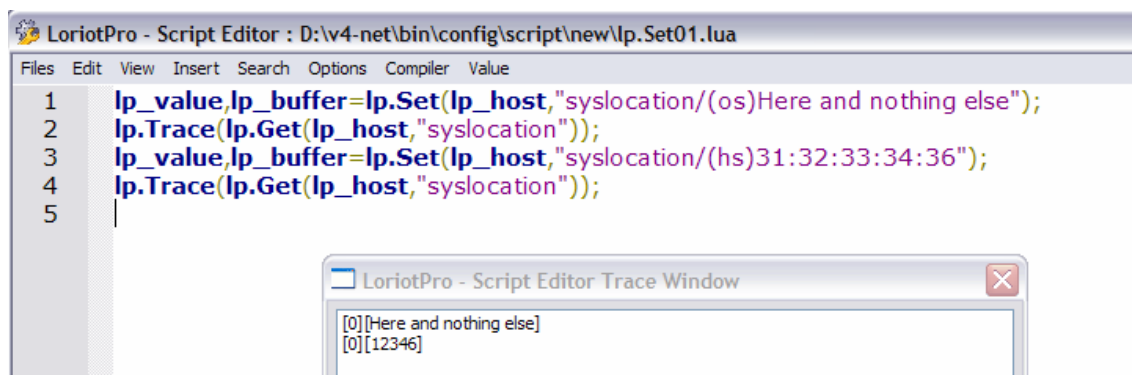
Renvoie **nil** si une erreur est rencontrée

Exemple :

```
lp_value,lp_buffer=lp.Set(lp_host,"syslocation/(os)Here and nothing else");  
lp.Trace(lp.Get(lp_host,"syslocation"));  
lp_value,lp_buffer=lp.Set(lp_host,"syslocation/(os)Here with us");  
lp.Trace(lp.Get(lp_host,"syslocation"));
```



```
lp_value,lp_buffer=lp.Set(lp_host,"syslocation/(os)Here and nothing else");  
lp.Trace(lp.Get(lp_host,"syslocation"));  
lp_value,lp_buffer=lp.Set(lp_host,"syslocation/(hs)31:32:33:34:36");  
lp.Trace(lp.Get(lp_host,"syslocation"));
```



Ip.SetBRC

```
value,buffer=Ip.SetBRC("ip","oid",vlanid);
```

Explication

Ip.SetBRC réalise une modification de l'objet SNMP unique « oid ». La fonction retourne deux valeurs ou nul si une erreur est trouvée.

Cette fonction permet de collecter des informations dans les instances de MIB par Vlan de Cisco.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x.

Attention

Cette adresse doit impérativement exister dans la Directory et la community SNMP RW de « ip » doit être configuré et valide.

« oid »

C'est une chaîne de caractères contenant un unique OID SNMP et un codage permettant de fournir la nouvelle valeur.

Oidname.index/(type)value

syntax	Value type
(nu)	NUL
(id)	Object ID
(hs)	Hexa string
(os)	Octet string
(in)	Integer
(ip)	IP adresse
(co)	Counter
(ga)	Gauge
(ti)	Timeticks

Attention

OIDname doit impérativement exister dans votre MIB database.

Vlanid le numero de VLAN a qui s'adresse votre requete

Paramètres en sortie

value :

Le résultat numérique si l'objet SNMP « oid » est de type integer/gauge... sinon 0 si il est de type octetstring.

Buffer :

Le résultat de la commande Set sous la forme d'une chaîne de caractères.

Renvoie **nil** si une erreur est rencontrée

buffer :

Contient sous la forme d'une chaîne de caractères le résultat du GET SNMP.

Renvoie **nil** si une erreur est rencontrée

Ip.GetNext

```
value,buffer,OIDName=Ip.GetNext("ip","oid");
```

Explication

Ip.GetNext réalise une interrogation SNMP de type GETNEXT d'un seul objet « oid » vers l'adresse « ip ». La fonction retourne trois valeurs :

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x.

Attention cette adresse doit impérativement exister dans la Directory.

« oid »

C'est une chaîne de caractères contenant un unique OID SNMP.

Attention cet OID doit impérativement exister dans votre database, il est possible d'utiliser une notation « dotted » .

Paramètres en sortie

value :

Le résultat numérique si l'objet SNMP « oid » est de type integer/gauge... sinon 0 si il est de type octetstring.

Renvoie **nil** si une erreur est rencontrée

buffer :

Contient sous la forme d'une chaîne de caractères le résultat du GETNEXT SNMP.

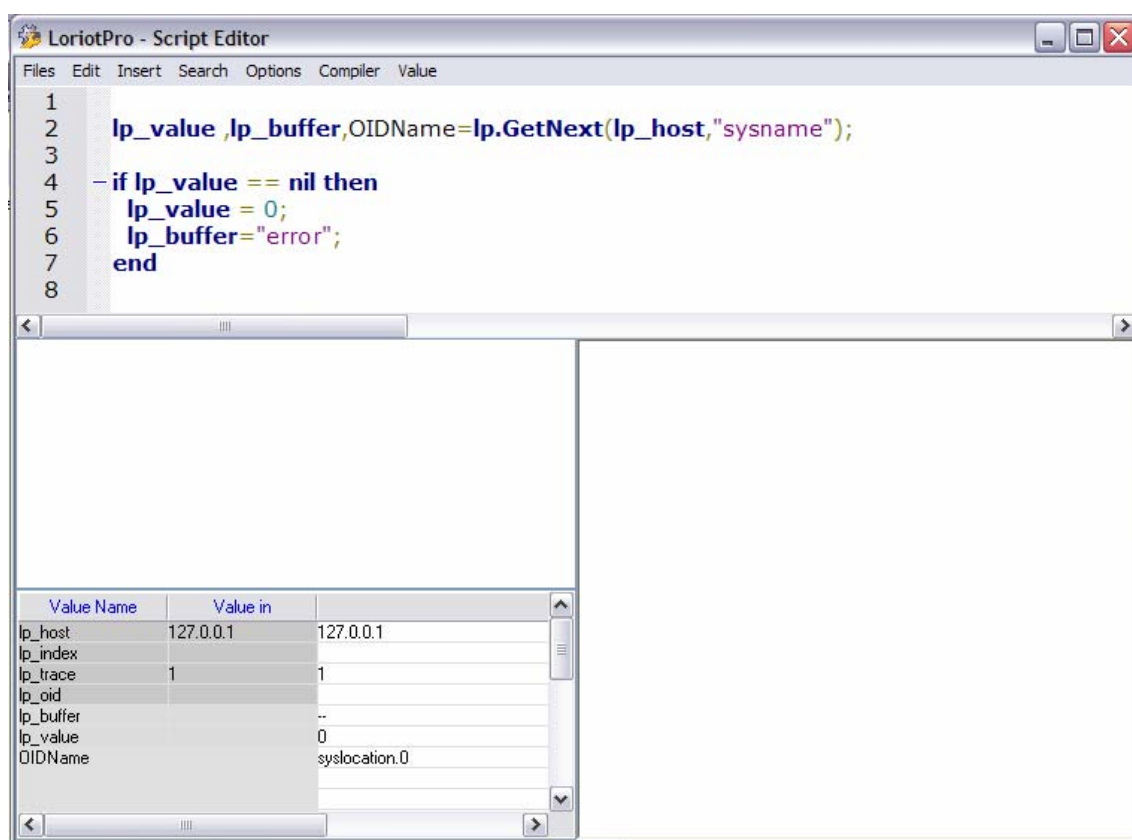
Renvoie **nil** si une erreur est rencontrée

OIDName :

Contient sous la forme d'une chaîne de caractères le nom de l'OID suivant retourné par le GETNEXT SNMP.

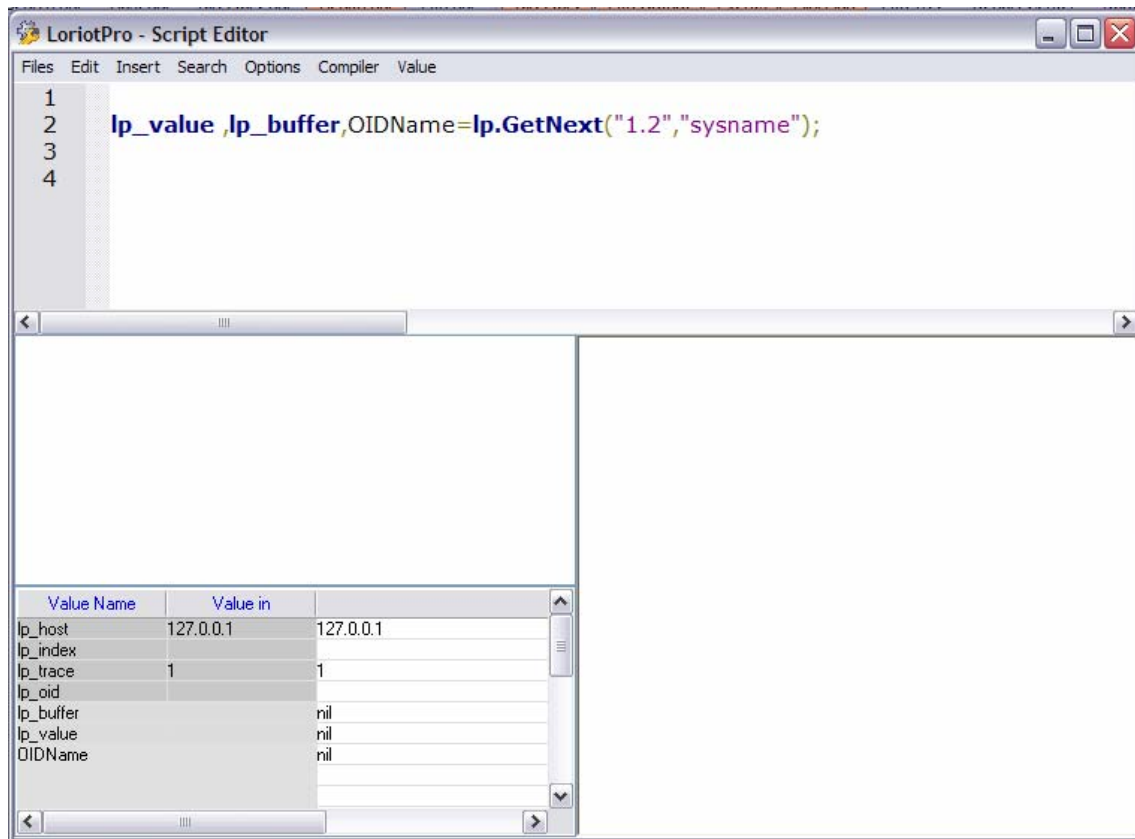
Renvoie **nil** si une erreur est rencontrée

Exemple



Dans cet exemple le script utilise la variable **lp_host** passé par LoriotPro au script et demande l'OID SNMP Next de *sysname* (*syslocation.0*).

Les résultats sont affichés dans la colonne « Value out ».



Dans cet exemple il y a une erreur de syntaxe au niveau du paramètre « ip » et toutes les valeurs en retour sont à **nil**.

```

OIDName="hrstoragedescr.0";
buffer = "";
ip_value = 0;
ip_buffer = "no found";
while ip_value do
    ip_value ,buffer,OIDName=Ip.GetNext(ip_host,OIDName);
    Ip.Trace(OIDName);
    if (string.sub(buffer,1,2) == "C:") then
        Ip.Trace(buffer);
        ip_buffer=buffer;
    end;
if (string.sub(OIDName,1,14) ~= "hrstoragedescr") then break end;
end

```


The screenshot shows the LoriotPro Script Editor with a Lua script and its execution results. The script is as follows:

```

1  OIDName="hrstoragedescr.0";
2  buffer = "";
3  lp_value = 0;
4  lp_buffer = "no found";
5  -- while lp_value do
6      lp_value ,buffer,OIDName=lp.GetNext(lp_host,OIDName);
7      lp.Trace(OIDName);
8      -- if (string.sub(buffer,1,2) == "C:" ) then
9          lp.Trace(buffer);
10         lp_buffer=buffer;
11     end;
12 if (string.sub(OIDName,1,14) ~= "hrstoragedescr") then break end;
13 end
    
```

The execution results are shown in a list:

```

[hrstoragedescr.1]
[hrstoragedescr.2]
[C:\Label:BOOT Serial Number c4cefa82]
[hrstoragedescr.3]
[hrstoragedescr.4]
[hrstoragedescr.5]
[hrstoragedescr.6]
[hrstoragedescr.7]
[hrstoragedescr.8]
[hrstoragedescr.9]
[hrstoragedescr.10]
[hrstoragedescr.11]
[hrstoragedescr.12]
[hrstorageallocationunits.1]
    
```

Below the list is a table showing the values of variables before and after execution:

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		C:\Label:BOOT Serial Number c4cefa82
lp_value		0
OIDName		hrstorageallocationunits.1

Un exemple plus complexe ou nous recherchons l'emplacement du disque C :

Ip.GetNextBRC

```
value,buffer,OIDName=Ip.GetNextBRC("ip","oid",vlanid);
```

Explication

Ip.GetNextBRC réalise une interrogation SNMP de type GETNEXT d'un seul objet « oid » vers l'adresse « ip ». La fonction retourne trois valeurs :
Cette fonction permet de collecter des informations dans les instances de MIB par Vlan de Cisco.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x.

Attention cette adresse doit impérativement exister dans la Directory.

« oid »

C'est une chaîne de caractères contenant un unique OID SNMP.

Attention cet OID doit impérativement exister dans votre database, il est possible d'utiliser une notation « dotted » .

Vlanid le numero de VLAN a qui s'adresse votre requete

Paramètres en sortie

value :

Le résultat numérique si l'objet SNMP « oid » est de type integer/gauge... sinon 0 si il est de type octetstring.

Renvoie **nil** si une erreur est rencontrée

buffer :

Contient sous la forme d'une chaîne de caractères le résultat du GETNEXT SNMP.

Renvoie **nil** si une erreur est rencontrée

OIDName :

Contient sous la forme d'une chaîne de caractères le nom de l'OID suivant retourné par le GETNEXT SNMP.

Renvoie **nil** si une erreur est rencontrée

Ip.Gets

```
reponsenumber =Ip.Gets("ip","oid1,oid2,...","arrayname");
```

Explication

Ip.Gets réalise une interrogation SNMP de type GET avec plusieurs objets « oid » vers l'adresse « ip ». La fonction retourne 1 valeurs et un tableau contenant les résultats de la requête. Cette fonction permet de collecter en une seule opération jusqu'à 15 objets OID sur un agent (« ip ») donnée.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x. Attention cette adresse doit impérativement exister dans la Directory.

« oid1,oid2 »

C'est une chaîne de caractères contenant une liste d'OID SNMP séparé par des virgules et sans aucun caractères blanc.

Attention ces OIDs doivent impérativement exister dans votre database, il est possible d'utiliser une notation « doted » .

Paramètres en sortie

reponsenumber:

le nombre de réponse sinon 0.

Renvoie **nil** si une erreur est rencontrée

arrayname :

Contient sous la forme d'un tableau l'ensemble des OID demandés.

Si la commande est Ip.Gets(lp_host, « sysname.0,syslocation.0 », « myarray ») ;

Il y a deux valeurs dans le tableau :

```
myarray["sysname.0"]  
myarray["syslocation.0"]
```

Renvoie **nil** si une erreur est rencontrée

Exemple

```

lp_value = 0;
lp_buffer = "no found";

lp_value=lp.Gets(lp_host,"sysname.0,syslocation.0", "a") ;

if lp_value ~= nil then
lp.Trace(a["sysname.0"]);
lp_buffer=a["sysname.0"];
lp.Trace(a["syslocation.0"]);
end

```

The screenshot shows the 'LoriotPro - Script Editor' window with the following code:

```

1  lp_value = 0;
2  lp_buffer = "no found";
3
4  lp_value=lp.Gets(lp_host,"sysname.0,syslocation.0", "a") ;
5
6  -if lp_value ~= nil then
7    lp.Trace(a["sysname.0"]);
8    lp_buffer=a["sysname.0"];
9    lp.Trace(a["syslocation.0"]);
10 end

```

The output window shows the execution results for the device [LUTEUS02]:

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		LUTEUS02
lp_value		2
		nil

Ip.GetsBRC

```
reponsenumber =Ip.GetsBRC("ip","oid1,oid2,...","arrayname",vlanid);
```

Explication

Ip.GetsBRC réalise une interrogation SNMP de type GET avec plusieurs objets « oid » vers l'adresse « ip ». La fonction retourne 1 valeurs et un tableau contenant les résultats de la requête. Cet fonction permet de collecter en une seule opération jusqu'à 15 objets OID sur un agent (« ip ») donnée.

Cette fonction permet de collecter des informations dans les instances de MIB par Vlan de Cisco.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x. Attention cette adresse doit impérativement exister dans la Directory.

« oid1,oid2 »

C'est une chaîne de caractères contenant une liste d'OID SNMP séparé par des virgules et sans aucun caractère blanc.

Attention ces OIDs doivent impérativement exister dans votre database, il est possible d'utiliser une notation « doted » .

Vlanid le numero de VLAN a qui s'adresse votre requête

Paramètres en sortie

reponsenumber:

le nombre de réponse sinon 0.

Renvoie **nil** si une erreur est rencontrée

arrayname :

Contient sous la forme d'un tableau l'ensemble des OID demandé.

Si la commande est Ip.Gets(lp_host, « sysname.0,syslocation.0 », « myarray ») ;

Il y a deux valeur dans le tableau :

```
myarray["sysname.0"]  
myarray["syslocation.0"]
```

Renvoie **nil** si une erreur est rencontrée

Ip.GetRows

```
linenumber, rownumber, indexnumber=Ip.GetRows("ip", "oid1,oid2", "arrayname");
```

Explication

Ip.GetRows réalise une interrogation SNMP de type GETNEXT avec plusieurs objets « oid » de type indexé vers l'adresse « ip ». La fonction retourne le nombre de ligne du tableau et un tableau contenant les résultats de la requête. Cette fonction permet de collecter en une seule opération jusqu'à 15 colonnes d'un tableau sur un agent (« ip ») donnée.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x. Attention cette adresse doit impérativement exister dans la Directory.

« oid1,oid2 »

C'est une chaîne de caractères contenant une liste d'OID SNMP séparés par des virgules et sans aucun caractère blanc.

Attention les oid doivent appartenir au même tableau et ces OIDs doivent impérativement exister dans votre database, il n'est pas possible d'utiliser une notation « dotted » .

Paramètres en sortie

Linenumber : Le nombre de ligne du tableau

rownumber: Le nombre **de colonne** du tableau collecté sinon 0.

Indexnumber : Le nombre d'index associé aux objets collecté

Renvoie **nil** si une erreur est rencontrée

arrayname :

Contient sous la forme d'un tableau l'ensemble des OID demandé.

Pour chaque ligne retournée on a une entrée **arrayname[l-%i]** ou %i est le numéro de la ligne, cette variable contient l'index utilisé pour la ligne.

Pour chaque valeur de la ligne (colonnes) on a une entrée de type **arrayname[oid-%i]** ou %i est le numéro de la ligne.

Renvoie **nil** si une erreur est rencontrée

V400 b138 SP0-bs 9 February 2006 :

Correct : lp.GetRows LUA function manage sparse table return "[shifted]" if the row filed is empty

Exemple

```
function PrintTable(entry)
a,b=lp.GetTableEntryList(entry,"array");
if a then
for j=0,(b-1) do
lp.Print("[",array["I-".j],"]");
end
lp.Print("\t");
hh="";
for j=0,(a-1) do
lp.Print(array[j],"\t");
hh=hh..array[j];
hh=hh..",";
end
lp.Print("\n");
c,d,e=lp.GetRows(lp_host,hh,"array2");
if c then
--print line
for j=0,(c-1) do
--lp.Print("[",array2["I-".j],"]","\t")
--print index
for l=0,(e-1) do
lp.Print("[",array2[string.format("OI-%i-%i",j,l)],"]","\t")
end
--print row value
for k=0,(d-1) do
lp.Print(array2[string.format("%s-%i",array[k],j)],"\t")
end
lp.Print("\n");
end
end
end
end
end
```

```

21 lp.Print(array[j],"\t");
22 hh=hh..array[j];
23 hh=hh..";";
24 end
25 lp.Print("\n");
26 c,d,e=lp.GetRows(lp_host,hh,"array2");
27 --if c then
28 --print line
29 -- for j=0,(c-1) do
30 --lp.Print("[",array2["1"..j],"]","\t")
31 --print index
32 -- for l=0,(e-1) do
33 lp.Print("[",array2[string.format("0l-%i-%i",j,l)],"]","\t")
34 end
35 --print row value
36 -- for k=0,(d-1) do
37 lp.Print(array2[string.format("%s-%i",array[k],j)],"\t")
38 end
39 lp.Print("\n");
40 end
41 end
42 end
43 end
44 --start program
45 PrintTable("ifentry")
46
47

```

```

File
1 [Ifindex] ifindex ifdescr iftype ifmtu ifspeed ifphysaddress ifadminstatus ifoperstatu
2 [1] 1 MS TCP Loopback interface[00] softwareLoopback 1520 10000000 -- up up
3 [2] 2 Broadcom 570x Gigabit Integrated Controller - ProtoWall Miniport[00] ethernetCsmacd
4 [3] 3 Dell TrueMobile 1300 WLAN Mini-PCI Card - Deterministic Network Enhancer Miniport[00] ether
5

```

Ip.GetRowsBRC

```
linenumber, rownumber, indexnumber  
=Ip.GetRows("ip", "oid1,oid2", "arrayname", vlanid);
```

Explication

Ip.GetRowsBRC réalise une interrogation SNMP de type GETNEXT avec plusieurs objets « oid » de type indexé vers l'adresse « ip ». La fonction retourne le nombre de lignes du tableau et un tableau contenant les résultats de la requête. Cette fonction permet de collecter en une seule opération jusqu'à 15 colonnes d'un tableau sur un agent (« ip ») donné.

Cette fonction permet de collecter des informations dans les instances de MIB par Vlan de Cisco.

Paramètres en entrée

« ip » :

C'est une chaîne de caractères contenant une adresse IP sous la forme x.x.x.x. Attention cette adresse doit impérativement exister dans la Directory.

« oid1,oid2 »

C'est une chaîne de caractères contenant une liste d'OID SNMP séparés par des virgules et sans aucun caractère blanc.

Attention les oid doivent appartenir au même tableau et ces OIDs doivent impérativement exister dans votre database, il n'est pas possible d'utiliser une notation « dotted » .

Vlanid le numero de VLAN a qui s'adresse votre requête

Paramètres en sortie

Linenumber : Le nombre de ligne du tableau

rownumber: Le nombre **de colonne** du tableau collecté sinon 0.

Indexnumber : Le nombre d'index associés aux objets collectés

Renvoie **nil** si une erreur est rencontrée

V400 b138 SP0-bs 9 February 2006 :

Correct : Ip.GetRowsBRC LUA function manage sparse table return "[shifted]" if the row filed is empty

arrayname :

Contient sous la forme d'un tableau l'ensemble des OID demandé.

Pour chaque ligne retourné on a une entrée ***arrayname[l-%i]*** ou %i est le numéro de la ligne, cette variable contient l'index utilisé pour la ligne.
Pour chaque valeur de la ligne (colonnes) on a une entrée de type ***arrayname[oid-%i]*** ou %i est le numéro de la ligne.

Renvoie **nil** si une erreur est rencontrée

Ip.GetPath

```
path=Ip.GetPath();
```

Explication

Ip.GetPath() retourne le path d'installation du logiciel, cela permet de travailler sur les répertoire de LoriotPro.

Paramètres en entrée

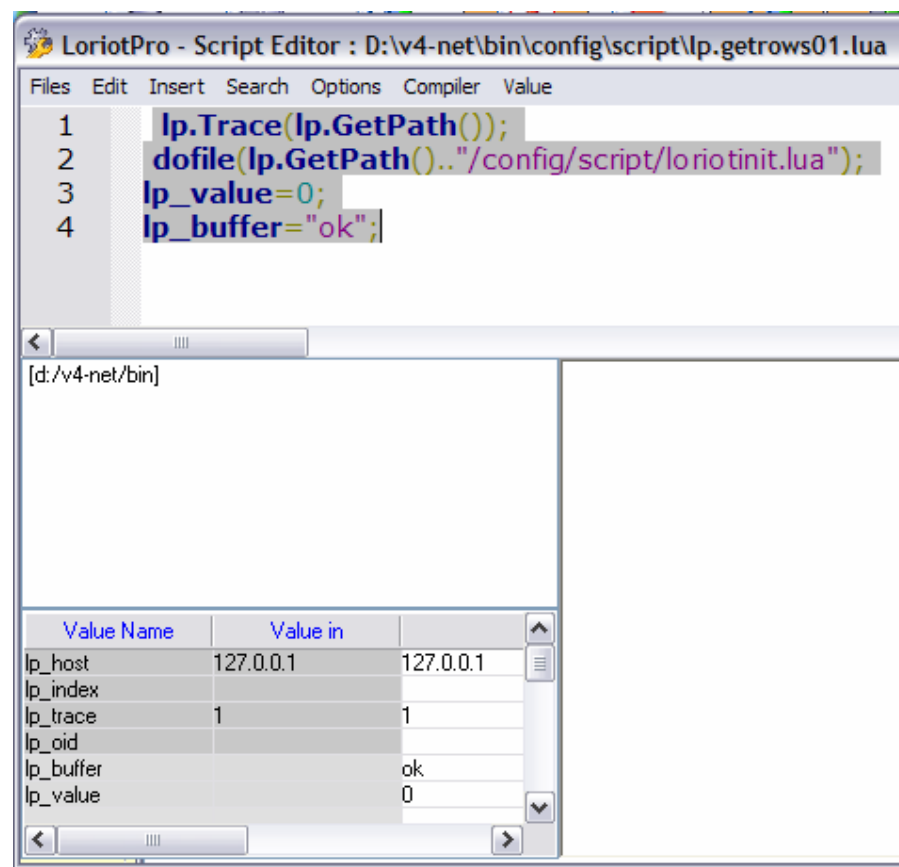
Paramètres en sortie

path:

Une chaîne de caractères contenant le path d'installation de LoriotPro.

Exemple

```
Ip.Trace(Ip.GetPath());  
dofile(Ip.GetPath().."/config/script/loriotinit.lua");  
lp_value=0;  
lp_buffer="ok";
```



Ip.Trace

```
Ip.Trace("string of char", "string",string);
```

Explication

Ip.Trace(« string of char ») permet d'afficher dans la zone système de l'éditeur de script des informations pour aider au « debugging ».

Attention

Ip.Trace ne fonctionne que si le script est lancé à travers une dialogue box et que la variable **Ip_trace=1**.

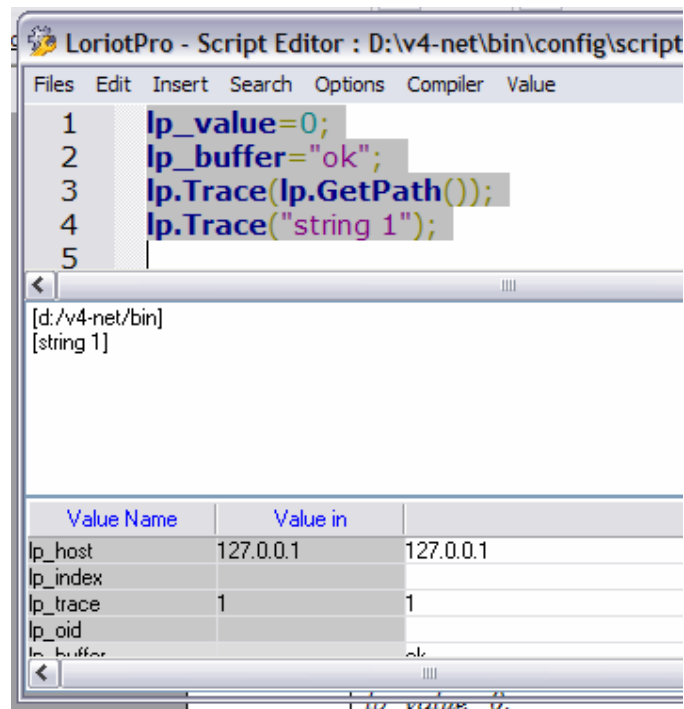
Paramètres en entrée

Une chaîne de caractères.

Paramètres en sortie

Exemple

```
Ip_value=0;  
Ip_buffer="ok";  
Ip.Trace(Ip.GetPath());  
Ip.Trace("string 1");
```



Ip.Print

```
Ip.Print("string of char", "string",string);
```

Explication

Ip.Print(« string of char ») permet d'afficher dans la zone système d'un script des informations.

Paramètres en entrée

Une chaîne de caractères.

Paramètres en sortie

Exemple

Ip.Break

Ip.Break("string of char");

Explication

Ip.Break(« string of char ») permet de poser un break dans un script pour aider au « debugging ».

Paramètres en entrée

Une chaîne de caractères qui sera affichée avec un « popup »

Paramètres en sortie

Un Popup

Attention, n'est utilisable que dans l'éditeur.

Exemple

```
Ip_value=0;
Ip_buffer="ok";
Ip.Break("pass 1");
Ip.Trace(Ip.GetPath());
Ip.Trace("string 1");
```

The screenshot shows the 'LoriotPro - Script Editor' window. The title bar indicates the file path: 'D:\v4-net\bin\config\script\lp.Trace01.lua'. The menu bar includes 'Files', 'Edit', 'Insert', 'Search', 'Options', 'Compiler', and 'Value'. The script content is as follows:

```
1  lp_value=0;  
2  lp_buffer="ok";  
3  lp.Break("pass 1");  
4  lp.Trace(lp.GetPath());  
5  lp.Trace("string 1");  
6  
7
```

A 'Break' dialog box is displayed in the center, asking 'Break at line [pass 1] Continue ?' with 'Oui' and 'Non' buttons. Below the script editor, a table shows variable values:

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		ok
lp_value		0

Ip.GetFirstIP

```
“ip”=Ip.GetFirstIP();
```

Explication

“**ip**”=**Ip.GetFirstIP()** permet de récupérer sous la forme d'une chaîne de caractère la première adresse IP de la directory . Cette fonction associée à d'autres permet de « browser » l'ensemble des host contenus dans la directory.

Paramètres en entrée

Paramètres en sortie

L'adresse IP du premier host de la directory sous la forme d'une chaîne de caractères ou **nil**

Exemple

```
Ip_value=0;
Ip_buffer="ok";
a=Ip.GetFirstIP();
if a~= nil then
    while a~= nil do
        Ip.Trace(a);
        a=Ip.GetNextIP(a);
    end
end
```

The screenshot shows the 'LoriotPro - Script Editor' window. The title bar indicates the file path is 'D:\v4-net\bin\config\scrip'. The menu bar includes 'Files', 'Edit', 'Insert', 'Search', 'Options', 'Compiler', and 'Value'. The script editor contains the following Lua code:

```
1  lp_value=0;
2  lp_buffer="ok";
3  a=lp.GetIPFirst();
4  if a~= nil then
5      while a~= nil do
6          lp.Trace(a);
7          a=lp.GetIPNext(a);
8      end
9  end
10
```

Below the script editor, the execution output is displayed in a scrollable area, showing the following IP addresses:

```
[82.65.181.65]
[192.168.1.200]
[192.168.1.100]
[192.168.1.105]
[192.168.1.1]
[127.0.0.1]
```

At the bottom of the window, there is a table with the following data:

Value Name	Value in	
lp_host	127.0.0.1	127.0.0.1
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

Ip.GetNextIP

```
“ip”=Ip.GetNextIP(« ip_old»);
```

Explication

“**ip**”=**Ip.GetNextIP()** permet de récupérer sous la forme d’une chaîne de caractères l’adresse IP suivante du host suivant dans la chaîne des hosts. Cette fonction permet associée à d’autres de « browser » l’ensemble des hosts contenus dans la directory.

Paramètres en entrée

Une adresse IP sous la forme d’une chaîne de caractères.

Paramètres en sortie

L’adresse IP du host suivant dans la « directory » sous la forme d’une chaîne de caractères ou **nil**

Exemple

Voir la commande “**ip**”=**Ip.GetFirstIP()**.

Ip.GetIPInformation

```
Result=Ip.GetIPInformation(« ip », "arrayname");
```

Explication

Result=Ip.GetIPInformation(« ip », "arrayname"); permet de récupérer sous la forme d'un tableau l'ensemble des paramètres associés à l'adresse « IP ».

Paramètres en entrée

Une adresse IP sous la forme d'une chaîne de caractères.

Un nom sous la forme d'une chaîne de caractères pour le tableau de stockage des réponses.

Paramètres en sortie

Result =1 ou nil si une erreur.

Si Result =1 : un tableau

Variables en retour	description
arrayname['ip']	L'adresse ip de host sous la forme d'une chaîne de caractères.
arrayname['ip_add']	L'adresse ip de host sous la forme d'un entier.
arrayname['name']	Le nom du host sous la forme d'une chaîne de caractères.
arrayname['status']	Le statut actuel du host (0 1 2 3 4)
arrayname['communityro']	La community SNMP Read Only du host
arrayname['communityrw']	La community SNMP Read Write du host
arrayname['user1']	Le paramètre user1 du host sous la forme d'une chaîne de caractères.
arrayname['user2']	Le paramètre user2 du host sous la forme d'une chaîne de caractères.
arrayname['user3']	Le paramètre user3 du host sous la forme d'une chaîne de caractères.
arrayname['sysid']	Le paramètre sysid du host sous la forme d'une chaîne de caractères.
arrayname['sysname']	Le paramètre sysname du host sous la forme d'une chaîne de caractères.
arrayname['isinrepair']	1 si le host est en période de maintenance sinon 0
arrayname['received']	Le nombre de trap reçu mémorisé.
arrayname['sla']	1 si le logiciel collecte les donnée SLA du host

arrayname['ri_ifgraph']	1 si le logiciel collecte des informations de bande passante sur cette adresse IP.
arrayname['ri_ifin octets_p']	Pourcentage d'utilisation en entré
arrayname['ri_ifout octets_p']	Pourcentage d'utilisation en sortie
arrayname['snmppolling']	1 si le polling snmp est activé
arrayname['pingpolling']	1 si le polling icmp est activé
arrayname['timesfirst']	Timestamp du dernier polling envoyé
arrayname['timeslast']	Timestamp du dernier polling reçu
arrayname['UID']	UID 64bits associé au host sous la forme d'une chaîne de caractères
arrayname['mac']	Retourne l'adresse MAC associé au host (si dans la base)
arrayname['ip_switch']	Retourne l'adresse IP du commutateur associé au host (si la valeur a été entrée (voir SetIPOption))
arrayname['ip_switch_index']	Retourne l'index du port du commutateur associé au host (si la valeur a été entrée (voir SetIPOption))
Arrayname['poll_interval']	L'intervalle de polling
Arrayname['number_status_4']	Le nombre de retransmission avant de passer en status 4
Arrayname['number_status_3']	Le nombre de retransmission avant de passer en status 3
Arrayname['isdynamic']	Ce host utilise un nomage dynamique 1 = actif 0 = non
Arrayname['dynamic_dns_name']	Le nom dns du host
Arrayname['repair_start']	Un timestamp de départ d'un période de maintenance
Arrayname['repair_end']	Un timestamp de fin d'un période de maintenance
Arrayname['repair_lock_polling']	La période de maintenance désactive le polling 1 = oui 0 = non
Arrayname['repair_lock_event_action']	La période de maintenance désactive les actions (filtre d'événement) associées à ce host 1 = oui 0 = non

V400 b138 SP0-bs 9 February 2006 :

Add : in Ip.GetIPInformations(ip,array) the mac value (array.mac) to get the saved mac add of the host.

V400 b138 SP0-bw 11 mars 2006 :

Include : to manage the host mobility monitoring.

LUA_IP_SWITCH and LUA_IP_SWITCH_INDEX in Ip.SetIpOption
array.ip_switch array.ip_switch_index in Ip.GetIPInformation

V400 b138 SP0-ch 10 Jun 2006 :

Add: lua option for Ip.GetIPInformation('ip','array');

array.poll_interval = polling interval in seconde
array.number_status_4 = number of retry before to pass to level 4
array.number_status_3 = number of retry before to pass to level 3
array.isdynamic = 1 si le host est géré par son nom DNS
array.dynamic_dns_name = le nom dns de la machine
array.repair_start = time stamp
array.repair_end = time stamp
array.repair_lock_polling = 0 / 1
array.repair_lock_event_action = 0 / 1

Exemple

```
Ip_value=0;
Ip_buffer="ok";
--a=Ip.GetFirstIP();
--if a~= nil then
    if Ip.GetIPInformation(Ip_host,"array") then
        Ip.Trace(array.name);
        Ip.Trace(array.ip);
        Ip.Trace(array.ip_add);
        Ip.Trace(array.name);
        Ip.Trace(array.status);
        Ip.Trace(array.communityro);
        Ip.Trace(array.communityrw);
        Ip.Trace(array.user1);
        Ip.Trace(array.user2);
        Ip.Trace(array.user3);
        Ip.Trace(array.sysid);
        Ip.Trace(array.sysname);
        Ip.Trace(array.isinrepair);
        Ip.Trace(array.trapreceived);
        Ip.Trace(array.sla);
        Ip.Trace(array.ri_ifgraph);
        Ip.Trace(array.ri_ifinotets_p);
        Ip.Trace(array.ri_ifoutoctets_p);
        Ip.Trace(array.snmpolling);
        Ip.Trace(array.pingpolling);
        Ip.Trace(array.timesfirst);
        Ip.Trace(array.timeslast);
    end
--end
```


The screenshot shows the 'LoriotPro - Script Editor' window. The script editor contains the following Lua code:

```

1  lp_value=0;
2  lp_buffer="ok";
3  --a=lp.GetIPFirst();
4  --if a~= nil then
5  -   if lp.GetIPInformation(lp_host,"array") then
6      lp.Trace(array.name);
7      lp.Trace(array.ip);
8      lp.Trace(array.ip_add);
9      lp.Trace(array.name);
10     lp.Trace(array.status);
11     lp.Trace(array.communityro);
12     lp.Trace(array.communityrw);

```

Below the script editor, the execution output is displayed in a scrollable area. It shows the following values:

```

[LoriotPro]
[127.0.0.1]
[16777343]
[LoriotPro]
[2]
[public]
[private]
[[null]]
[[null]]
[[null]]
[[null]]
[[null]]
[[null]]
[0]

[1]
[0]
[0]
[0]

```

At the bottom of the screenshot, there is a table with the following data:

Value Name	Value in	
lp_host	127.0.0.1	127.0.0.1
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

```

lp_value=0;
lp_buffer="ok";
if lp.GetIPInformation(lp_host,"array") then
  if array.status==2 then
    lp.Trace("host polled in snmp ", lp.Get(lp_host,"sysname"))
  end;
end

```

The screenshot shows the 'LoriotPro - Script Editor' window. The title bar indicates the file path: 'D:\v4-net\bin\config\script\lp.GetIPInformation01.lua'. The menu bar includes 'Files', 'Edit', 'Insert', 'Search', 'Options', 'Compiler', and 'Value'. The script content is as follows:

```
1 lp_value=0;  
2 lp_buffer="ok";  
3 - if lp.GetIPInformation(lp_host,"array") then  
4 if array.status==2 then lp.Trace("host polled in snmp ",lp.Get(lp_host,"sysname")) end;  
5 end  
6
```

Below the script editor, the execution output is displayed: '[host polled in snmp][0][LUTEUS02]'. At the bottom, a table shows the state of variables:

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		ok
lp_value		0

Ip.SetIPOption

```
result=Ip.SetIPOption('ip',param_type,"value");
```

Explication

Result=Ip.SetIPOption(« ip »,param_type, « value »); permet de modifier les paramètres système d'un host « ip ».

Paramètres en entrée

« ip » Une adresse IP sous la forme d'une chaîne de caractères.
 param_type L'identifiant du paramètre à modifier, la liste des valeurs est contenue dans le fichier *bin/config/script/loriotinit.lua*.

Qui peut être intégré au script avec la commande :

```
dofile(Ip.GetPath().."/config/script/loriotinit.lua");
```

Placé en début de script.

Paramètre	Valeur possible
LP_LUA_NAME=1	string
LP_LUA_COMMUNITYRO=2	string
LP_LUA_COMMUNITYRW=3	string
LP_LUA_USER1=4	string
LP_LUA_USER2=5	string
LP_LUA_USER3=6	string
LP_LUA_SYSID=7	string
LP_LUA_SYSNAME=8	string
LP_LUA_GROUPEF=9	« ip »
LP_LUA_POLLING_SNMP=10	« 0 » off « 1 » active
LP_LUA_POLLING_PING=11	« 0 » off « 1 » active
LP_LUA_ISDEVICE	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_MAC	String L'adresse MAC associé au host dans le format : "xx:xx:xx:xx:xx:xx:"
LP_IP_SWITCH	Permet de sauvegarder l'add IP du commutateur associé au host
LP_IP_SWITCH_INDEX	Permet de sauvegarder l'index du port sur le commutateur associé au host
LP_LUA_POLLING	Polling en seconde
LP_LUA_NUMBER_STATUS_3	Nombre de retransmissions avant de passer en status 3 Doit être inférieur à status 4

LP_LUA_NUMBER_STATUS_4	Nombre de retransmissions avant de passer en status 3 Doit être supérieur à status 3
------------------------	---

« value » La nouvelle valeur sous la forme d'une chaîne de caractères du paramètre.

```
V400 b138 SP0-bs 9 February 2006 :
Add ; Ip.SetIpOption(ip,LP_LUA_MAC,"xx:xx:xx:xx:xx:xx");

V400 b138 SP0-bw 11 mars 2006 :
Include : to manage the host mobility monitoring.
        LUA_IP_SWITCH and LUA_IP_SWITCH_INDEX in Ip.SetIpOption

V400 b138 SP0-ch 10 Jun 2006 :
Add : lua option for Ip.SetIpOption('ip',REF,value);
        REF = LP_LUA_POLLING
              polling in second
        REF = LP_LUA_NUMBER_STATUS_3
              number of retry before to pass in status 3
        REF = LP_LUA_NUMBER_STATUS_4
              number of retry before to pass in status 4
```

Paramètres en sortie

Result =1 ou nil si une erreur.

Exemple

```
lp_value=0;
lp_buffer="ok";
dofile(lp.GetPath("../config/script/loriotinit.lua");

    if lp.GetIPInformation(lp_host,"array") then
        lp.Trace(array.name);
    end
    lp.SetIpOption(lp_host,LP_LUA_NAME,"bob");
    if lp.GetIPInformation(lp_host,"array") then
        lp.Trace(array.name);
    end
```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\lp.SetIPOption01.lua

Files Edit Insert Search Options Compiler Value

```

1  lp_value=0;
2  lp_buffer="ok";
3  dofile(lp.GetPath().."/config/script/loriotinit.lua");
4
5  -   if lp.GetIPInformation(lp_host,"array") then
6      lp.Trace(array.name);
7  end
8  lp.SetIPOption(lp_host,LP_LUA_NAME,"bob");
9  -   if lp.GetIPInformation(lp_host,"array") then
10     lp.Trace(array.name);
11 end
12

```

[Linksys WAG54G]
[bob]

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

Ip.InsertNewContainer

```
“uid”=Ip.InsertNewContainer(“name”,“uid”,type);
```

Explication

Cette fonction permet d’insérer un container dans la directory à partir de l’objet « uid » d’insertion.

Paramètres en entrée

« name » Un nom sous la forme d’une chaîne de caractère pour le nom du container.
Attention de créer des noms de container unique pour une recherche plus facile.

« uid » Une valeur sur 64Bits sous la forme d’une chaîne de caractères qui est le référentiel d’un objet de la directory . « 0 » si inséré à partir du root.

Type le type de container à insérer.

type
LP_LUA_DT_COUNTRY=1; LP_LUA_DT_ORGANIZATION =2; LP_LUA_DT_ORGANIZATION_UNIT=3;

Paramètres en sortie

« uid » l’uid de l’objet container créé
nil si une erreur.

Ip.InsertNewNetwork

```
“uid”=Ip.InsertNewNetwork(“name”,“description”,“net”,“mask”,type,speed,“uid”);
```

Explication

Cette fonction permet d’insérer un nouveau network dans la directory à partir de l’objet « uid » d’insertion.

Attention

Logiquement les networks sont automatiquement créé par le module de découverte de LoriotPro, la création manuelle n’est pas toujours souhaitable. Si le network existe il ne sera pas recréé et la fonction retournera nil.

Paramètres en entrée

« name » Un nom sous la forme d’une chaîne de caractères pour le nom du network.

Attention de créer des noms de container uniques pour une recherche plus facile.

« description » la description du network

« net » le network sous la forme « x.x.x.x » ex : « 10.0.0.0 »

« mask » le mask du network sous le forme « x.x.x.x » ex : « 255.0.0.0 »

type le type de media du network (voir la rfc1213).

```
Loriottestentry ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "test snmpv2 ASN Mib."
    SYNTAX INTEGER {
        other(1),          -- none of the following
        regular1822(2),
        hdh1822(3),
        ddnX25(4),
        rfc877x25(5),
        ethernetCsmacd(6),
        iso88023Csmacd(7),
        iso88024TokenBus(8),
        iso88025TokenRing(9),
        iso88026Man(10),
        starLan(11),
        proteon10Mbit(12),
        proteon80Mbit(13),
        hyperchannel(14),
        fddi(15),
        lapb(16),
        sdlc(17),
        ds1(18),          -- DS1-MIB
        e1(19),          -- Obsolete see DS1-MIB
        basicISDN(20),
        primaryISDN(21),
        propPointToPointSerial(22), -- proprietary serial
        ppp(23),
        softwareLoopback(24),
        eon(25),          -- CLNP over IP
        ethernet3Mbit(26),
        nsip(27),        -- XNS over IP
        slip(28),        -- generic SLIP
        ultra(29),       -- ULTRA technologies
        ds3(30),         -- DS3-MIB
        sip(31),         -- SMDS, coffee
```

```

frameRelay(32),      -- DTE only.
rs232(33),
para(34),           -- parallel-port
arcnet(35),         -- arcnet
arcnetPlus(36),    -- arcnet plus
atm(37),            -- ATM cells
miox25(38),
sonet(39),          -- SONET or SDH
x25ple(40),
iso88022llc(41),
localTalk(42),
smdsDxi(43),
frameRelayService(44), -- FRNETSERV-MIB
v35(45),
hssi(46),
hippi(47),
modem(48),         -- Generic modem
aal5(49),          -- AAL5 over ATM
sonetPath(50),
sonetVT(51),
smdsIcip(52),      -- SMDS InterCarrier Interface
propVirtual(53),   -- proprietary virtual/internal
propMultiplexor(54), -- proprietary multiplexing
ieee80212(55),     -- 100BaseVG
fibreChannel(56), -- Fibre Channel
hippiInterface(57), -- HIPPI interfaces
frameRelayInterconnect(58), -- Obsolete use either
-- frameRelay(32) or
-- frameRelayService(44).
aflane8023(59),    -- ATM Emulated LAN for 802.3
aflane8025(60),    -- ATM Emulated LAN for 802.5
cctEmul(61),      -- ATM Emulated circuit
fastEther(62),    -- Fast Ethernet (100BaseT)
isdn(63),         -- ISDN and X.25
v11(64),          -- CCITT V.11/X.21
v36(65),          -- CCITT V.36
g703at64k(66),    -- CCITT G703 at 64Kbps
g703at2mb(67),    -- Obsolete see DS1-MIB
qllc(68),         -- SNA QLLC
fastEtherFX(69),  -- Fast Ethernet (100BaseFX)
channel(70),      -- channel
ieee80211(71),    -- radio spread spectrum
ibm370parChan(72), -- IBM System 360/370 OEMI Channel
escon(73),        -- IBM Enterprise Systems Connection
dlsW(74),         -- Data Link Switching
isdns(75),        -- ISDN S/T interface
isdnu(76),        -- ISDN U interface
lapd(77),         -- Link Access Protocol D
ipSwitch(78),     -- IP Switching Objects
rsrb(79),         -- Remote Source Route Bridging
atmLogical(80),   -- ATM Logical Port
ds0(81),          -- Digital Signal Level 0
ds0Bundle(82),    -- group of ds0s on the same ds1
bsc(83),          -- Bisynchronous Protocol
async(84),        -- Asynchronous Protocol
cnr(85),          -- Combat Net Radio
iso88025Dtr(86),  -- ISO 802.5r DTR
eplrs(87),        -- Ext Pos Loc Report Sys
arap(88),         -- Appletalk Remote Access Protocol
propCnls(89),     -- Proprietary Connectionless Protocol
hostPad(90),      -- CCITT-ITU X.29 PAD Protocol
termPad(91),      -- CCITT-ITU X.3 PAD Facility
frameRelayMPI(92), -- Multiproto Interconnect over FR
x213(93),         -- CCITT-ITU X213
adsl(94),         -- Asymmetric Digital Subscriber Loop
radsl(95),        -- Rate-Adapt. Digital Subscriber Loop
sdsl(96),         -- Symmetric Digital Subscriber Loop
vdsl(97),         -- Very H-Speed Digital Subscrib. Loop
iso88025CRFPInt(98), -- ISO 802.5 CRFP
myrinet(99),      -- Myricom Myrinet
voiceEM(100),     -- voice recEive and transMit
voiceFXO(101),    -- voice Foreign Exchange Office
voiceFXS(102),    -- voice Foreign Exchange Station
voiceEncap(103),  -- voice encapsulation
voiceOverIp(104), -- voice over IP encapsulation
atmDxi(105),      -- ATM DXI
atmFuni(106),     -- ATM FUNI

```



```

atmIma (107),          -- ATM IMA
pppMultilinkBundle(108), -- PPP Multilink Bundle
ipOverCdlc (109),     -- IBM ipOverCdlc
ipOverClaw (110),     -- IBM Common Link Access to Workstn
stackToStack (111),  -- IBM stackToStack
virtualIpAddress (112), -- IBM VIPA
mpc (113),            -- IBM multi-protocol channel support
ipOverAtm (114),     -- IBM ipOverAtm
iso88025Fiber (115), -- ISO 802.5j Fiber Token Ring
tdlc (116),          -- IBM twinaxial data link control
gigabitEthernet (117), -- Gigabit Ethernet
hdlc (118),          -- HDLC
lapf (119),          -- LAP F
v37 (120),           -- V.37
x25mlp (121),        -- Multi-Link Protocol
x25huntGroup (122),  -- X25 Hunt Group
trasnpHdlc (123),    -- Transp HDLC
interleave (124),    -- Interleave channel
fast (125),           -- Fast channel
ip (126),             -- IP (for APPN HPR in IP networks)
docsCableMaclayer (127), -- CATV Mac Layer
docsCableDownstream (128), -- CATV Downstream interface
docsCableUpstream (129), -- CATV Upstream interface
a12MppSwitch (130),  -- Avalon Parallel Processor
tunnel (131),         -- Encapsulation interface
coffee (132),        -- coffee pot
ces (133),            -- Circuit Emulation Service
atmSubInterface (134), -- ATM Sub Interface
l2vlan (135),         -- Layer 2 Virtual LAN using 802.1Q
l3ipvlan (136),       -- Layer 3 Virtual LAN using IP
l3ipxvlan (137),      -- Layer 3 Virtual LAN using IPX
digitalPowerline (138), -- IP over Power Lines
mediaMailOverIp (139), -- Multimedia Mail over IP
dtm (140),            -- Dynamic synchronous Transfer Mode
dcn (141),            -- Data Communications Network
ipForward (142),     -- IP Forwarding Interface
msdsl (143),          -- Multi-rate Symmetric DSL
ieee1394 (144),      -- IEEE1394 High Performance Serial Bus
if-gsn (145),         -- HIPPI-6400
dvbRccMacLayer (146), -- DVB-RCC MAC Layer
dvbRccDownstream (147), -- DVB-RCC Downstream Channel
dvbRccUpstream (148), -- DVB-RCC Upstream Channel
atmVirtual (149),    -- ATM Virtual Interface
mplsTunnel (150),    -- MPLS Tunnel Virtual Interface
srp (151),            -- Spatial Reuse Protocol
voiceOverAtm (152),  -- Voice Over ATM
voiceOverFrameRelay (153), -- Voice Over Frame Relay
idsl (154),           -- Digital Subscriber Loop over ISDN
compositeLink (155), -- Avici Composite Link Interface
ss7SigLink (156),    -- SS7 Signaling Link
propWirelessP2P (157), -- Prop. P2P wireless interface
frForward (158),     -- Frame Forward Interface
rfc1483 (159),       -- Multiprotocol over ATM AAL5
usb (160),            -- USB Interface
ieee8023adLag (161), -- IEEE 802.3ad Link Aggregate
bgppolicyaccounting (162), -- BGP Policy Accounting
frf16MfrBundle (163), -- FRF .16 Multilink Frame Relay
h323Gatekeeper (164), -- H323 Gatekeeper
h323Proxy (165),     -- H323 Voice and Video Proxy
mpls (166),           -- MPLS
mfSigLink (167),     -- Multi-frequency signaling link
hdlsl2 (168),        -- High Bit-Rate DSL - 2nd generation
shdsl (169),         -- Multirate HDSL2
dslFDL (170),        -- Facility Data Link 4Kbps on a DS1
pos (171),           -- Packet over SONET/SDH Interface
dvbAsiIn (172),     -- DVB-ASI Input
dvbAsiOut (173),    -- DVB-ASI Output
plc (174),           -- Power Line Communications
nfas (175),          -- Non Facility Associated Signaling
tr008 (176),         -- TR008
gr303RDT (177),     -- Remote Digital Terminal
gr303IDT (178),     -- Integrated Digital Terminal
isup (179),          -- ISUP
propDocsWirelessMaclayer (180), -- prop/Maclayer
propDocsWirelessDownstream (181), -- prop/Downstream
propDocsWirelessUpstream (182), -- prop/Upstream
hiperlan2 (183),    -- HIPERLAN Type 2 Radio Interface

```

```
propBWApt2Mp (184), -- PropBroadbandWirelessAccesspt2multipt
sonetOverheadChannel (185), -- SONEt Overhead Channel
digitalWrapperOverheadChannel (186), -- Digital Wrapper
aal2 (187), -- ATM adaptation layer 2
radioMAC (188), -- MAC layer over radio links
atmRadio (189) -- ATM over radio links
}
```

Speed la vitesse du media

« uid » Une valeur sur 64Bits sous la forme d'une chaîne de caractères qui est le référentiel d'un objet de la directory . « 0 » si inséré à partir du root.

Paramètres en sortie

« uid » l'uid de l'objet network créé
nil si une erreur.

Exemple

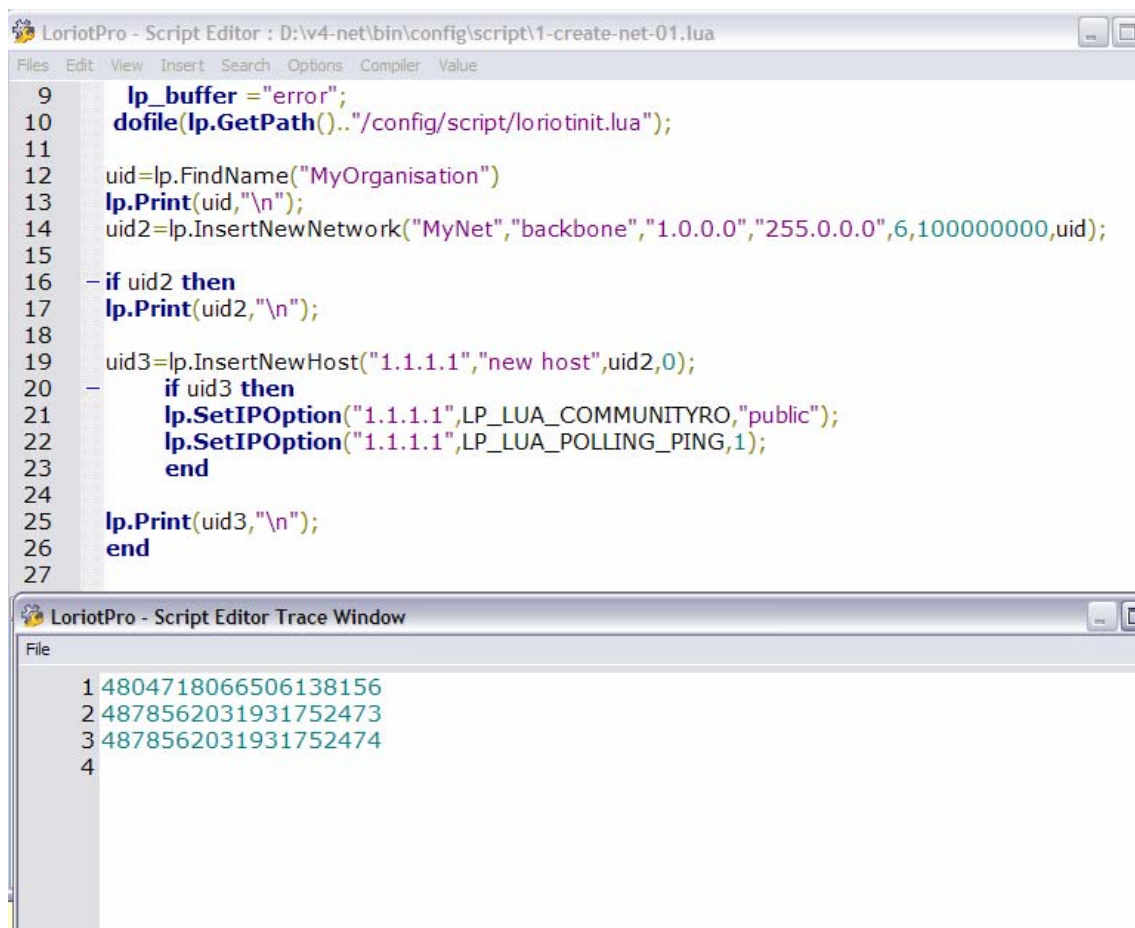
```
lp_value = 0;
lp_buffer ="error";
dofile(lp.GetPath().."/config/script/loriotinit.lua");

uid=lp.FindName("MyOrganisation")
lp.Print(uid,"\n");
uid2=lp.InsertNewNetwork("MyNet","backbone","1.0.0.0","255.0.0.0",6,100000000,uid);

if uid2 then
lp.Print(uid2,"\n");

uid3=lp.InsertNewHost("1.1.1.1","new host",uid2,0);
  if uid3 then
  lp.SetIPOption("1.1.1.1",LP_LUA_COMMUNITYRO,"public");
  lp.SetIPOption("1.1.1.1",LP_LUA_POLLING_PING,1);
  end

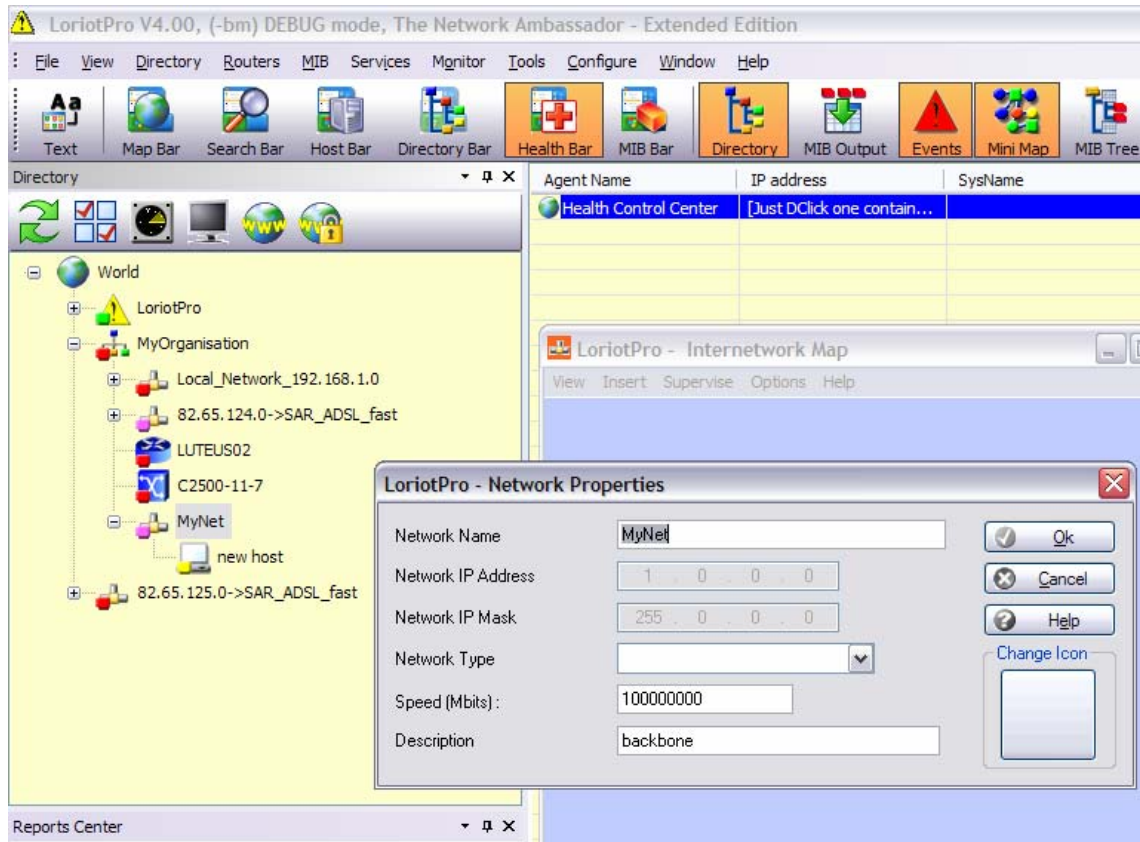
lp.Print(uid3,"\n");
end
```



The image shows two windows from the LoriotPro software. The top window is the 'Script Editor' for a file named '1-create-net-01.lua'. It contains a Lua script that performs several network configuration tasks: it sets an error buffer, finds an organization name, creates a new network named 'backbone' with IP ranges '1.0.0.0' to '255.0.0.0', inserts a new host with IP '1.1.1.1', and sets specific IP options for that host. The bottom window is the 'Script Editor Trace Window', which displays the output of the script's print statements, showing three lines of unique identifiers.

```
LoriotPro - Script Editor : D:\v4-net\bin\config\script\1-create-net-01.lua
Files Edit View Insert Search Options Compiler Value
9      lp_buffer = "error";
10     dofile(lp.GetPath().."/config/script/loriotinit.lua");
11
12     uid=lp.FindName("MyOrganisation")
13     lp.Print(uid,"\n");
14     uid2=lp.InsertNewNetwork("MyNet", "backbone", "1.0.0.0", "255.0.0.0", 6, 100000000, uid);
15
16     -if uid2 then
17     lp.Print(uid2, "\n");
18
19     uid3=lp.InsertNewHost("1.1.1.1", "new host", uid2, 0);
20     -if uid3 then
21     lp.SetIPOption("1.1.1.1", LP_LUA_COMMUNITYRO, "public");
22     lp.SetIPOption("1.1.1.1", LP_LUA_POLLING_PING, 1);
23     end
24
25     lp.Print(uid3, "\n");
26     end
27

LoriotPro - Script Editor Trace Window
File
1 4804718066506138156
2 4878562031931752473
3 4878562031931752474
4
```



Ip.InsertNewHost

```
“uid”=Ip.InsertNewHost(“ip”,“name”,“uid”,putinnetwork);
```

Explication

Cette fonction permet d’insérer un host dans la directory à partir de l’objet « uid » d’insertion.

Paramètres en entrée

« ip » l’adresse IP du host sous la forme « x.x.x.x »
« name » Un nom sous la forme d’une chaîne de caractère pour le nom du host
« uid » Une valeur sur 64Bits sous la forme d’une chaîne de caractères qui est le référentiel d’un objet de la directory . « 0 » si inséré à partir du root.
Putinnetwork Si 0 le host est inséré sous l’« uid ». Si 1 si un network existe dans la directory pour cette host il sera inséré en dessous du container network trouvé.

Paramètres en sortie

« uid » l’uid de l’objet host créé
nil si une erreur.

Exemple

```
-- Loriotpro V4
-- To run correctly this file is located to bin/config/script
-- Input values
-- Ip_index index for this script ".1"
-- Ip_oid SNMP OID for this script "ifnumber"
-- Ip_host default ip address for this script "127.0.0.1"
-- Output Values
  Ip_value = 0;
  Ip_buffer ="error";
dofile(Ip.GetPath().."/config/script/loriotinit.lua");

uid=Ip.FindName("MyOrganisation")
Ip.Print(uid,"\n");
uid2=Ip.InsertNewContainer("Container 1",uid,2);

if uid2 then
Ip.Print(uid2,"\n");

uid3=Ip.InsertNewHost("1.1.1.1","new host",uid2,0);
  if uid3 then
    Ip.SetIPOption("1.1.1.1",LP_LUA_COMMUNITYRO,"public");
    Ip.SetIPOption("1.1.1.1",LP_LUA_POLLING_PING,1);
  end
```

```
lp.Print(uid3,"n");
end
```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\1-create-01.lua

Files Edit View Insert Search Options Compiler Value

```

9      lp_buffer = "error";
10     dofile(lp.GetPath().."/config/script/loriotinit.lua");
11
12     uid=lp.FindName("MyOrganisation")
13     lp.Print(uid,"n");
14     uid2=lp.InsertNewContainer("Container 1",uid,2);
15
16     -if uid2 then
17     lp.Print(uid2,"n");
18
19     uid3=lp.InsertNewHost("1.1.1.1","new host",uid2,0);
20     -    if uid3 then
21     lp.SetIPOption("1.1.1.1",LP_LUA_COMMUNITYRO,"public");
22     lp.SetIPOption("1.1.1.1",LP_LUA_POLLING_PING,1);
23     end
24
25     lp.Print(uid3,"n");
26     end
27

```

Value Name	Value in	Value out
lp_host	82.65.125.164	82.65.125.164
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		error

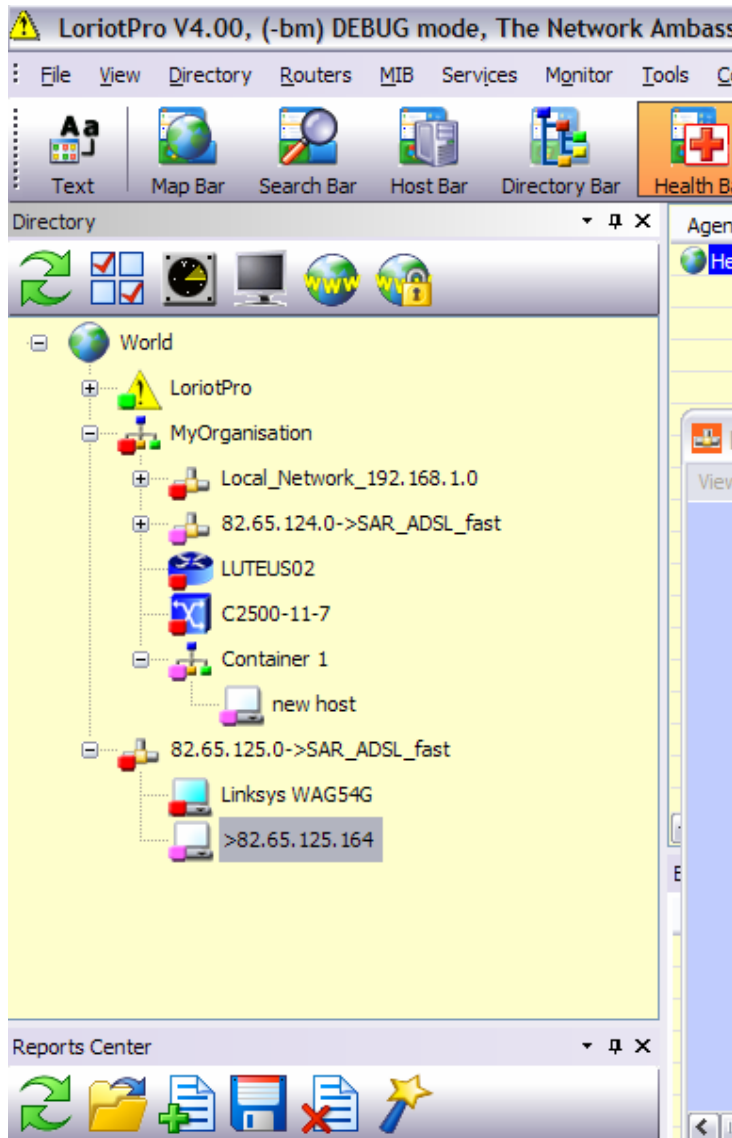
lp_ LoriotPro - Script Editor Trace Window

File

```

1 4804718066506138156
2 4878554721897414681
3 4878554721897414682
4

```



Ip.GetUIDInformation

```
Result=Ip.GetUIDInformation(uid,"arrayname");
```

Explication

Result=Ip.GetUIDInformation(uid,"arrayname"); permet de récupérer sous la forme d'un tableau l'ensemble des paramètres associés à un UID. Chaque élément de la directory possède un UID unique.

Paramètres en entrée

« Uid » Une valeur sur 64Bits sous la forme d'une chaîne de caractères qui est le référentiel d'un objet de la directory .
 « arrayname » Un nom sous la forme d'une chaîne de caractère pour le tableau de stockage des réponses.

Paramètres en sortie

Result =1 ou nil si une erreur.

Si Result =1 : un tableau

Variables en retour	description
arrayname['name']	Le nom de l'objet sous la forme d'une chaîne de caractère.
arrayname['status']	Le statut actuel de l'objet (0 1 2 3 4)
arrayname['plugin_name']	Le nom associé à un plugin sous la forme d'une chaîne de caractère si l'objet est un plugin.
arrayname['object_type']	<p>Le type d'objet (déclaré dans le fichier bin/config/script/loriotinit.lua.):</p> <p>Qui peut être intégré au script avec la commande :</p> <pre>dofile(lp.GetPath().."/config/script/loriotinit.lua");</pre> <pre>--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;</pre>
arrayname['string']	Le paramètre string de l'objet sous la forme d'une chaîne de caractère.

Exemple

```

lp_value = 0;
lp_buffer = "error";
dofile(lp.GetPath().."/config/script/loriotinit.lua");
lp_value=lp.GetIPInformation(lp_host,"a");
if lp_value~=nil then
    lp_value=lp.GetUIDInformation(a.UID,"array");
    if lp_value~=nil then
        lp.Trace("name =",array.name);
        lp.Trace("status =",array.status);
        lp.Trace("objet type =",array.object_type);
        if array.objet_type==LP_LUA_DT_HOST_PLUGIN then
            lp.Trace("plugin name =",array.plugin_name);
        end
        lp_buffer = "ok";
    end
end
end
end

```

The screenshot shows the LoriotPro Script Editor with the following Lua script:

```

1  lp_value = 0;
2  lp_buffer = "error";
3  dofile(lp.GetPath().."/config/script/loriotinit.lua");
4  lp_value=lp.GetIPInformation(lp_host,"a");
5  -if lp_value~=nil then
6      lp_value=lp.GetUIDInformation(a.UID,"array");
7      - if lp_value~=nil then
8          lp.Trace("name =",array.name);
9          lp.Trace("status =",array.status);
10         lp.Trace("objet type =",array.object_type);
11         - if array.objet_type==LP_LUA_DT_HOST_PLUGIN then
12             lp.Trace("plugin name =",array.plugin_name);
13         end
14         lp_buffer = "ok";
15     end
16 end
17

```

A Trace Window is open, displaying the following output:

```

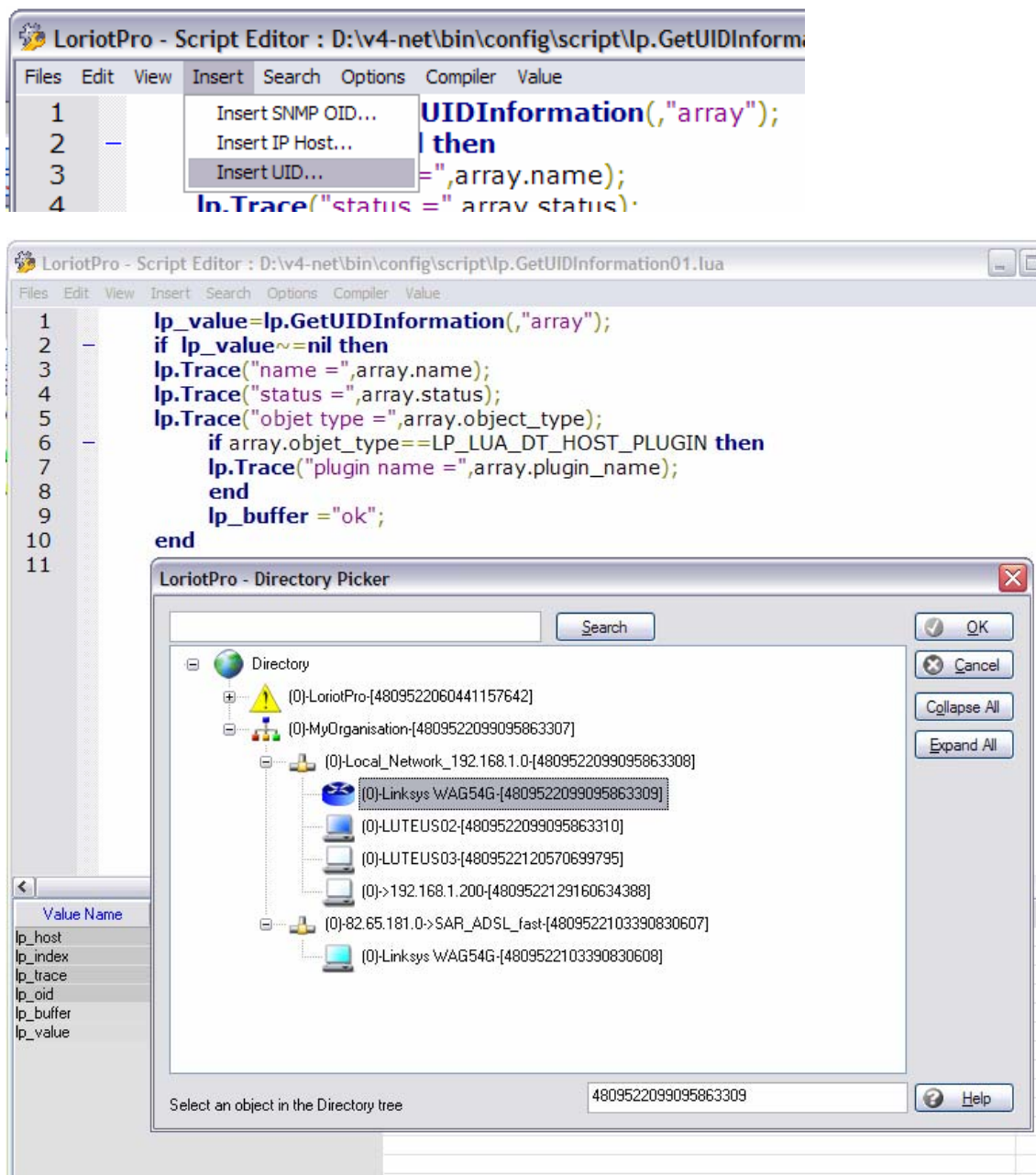
[name =][Linksys WAG54G]
[status =][4]
[objet type =][1]

```

At the bottom, a table shows the variable values:

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		1

Un « Wizard » intégré à l'éditeur permet de récupérer l'UID des objets de la Directory. L'insertion se fait à l'emplacement du curseur.



```

lp_value = 0;
lp_buffer ="error";

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

    lp_value=lp.GetUIDInformation("4809522060441157642", "array");
    if lp_value~=nil then
        lp.Trace("name =",array.name);
        lp.Trace("status =",array.status);
        lp.Trace("objet type =",array.object_type);

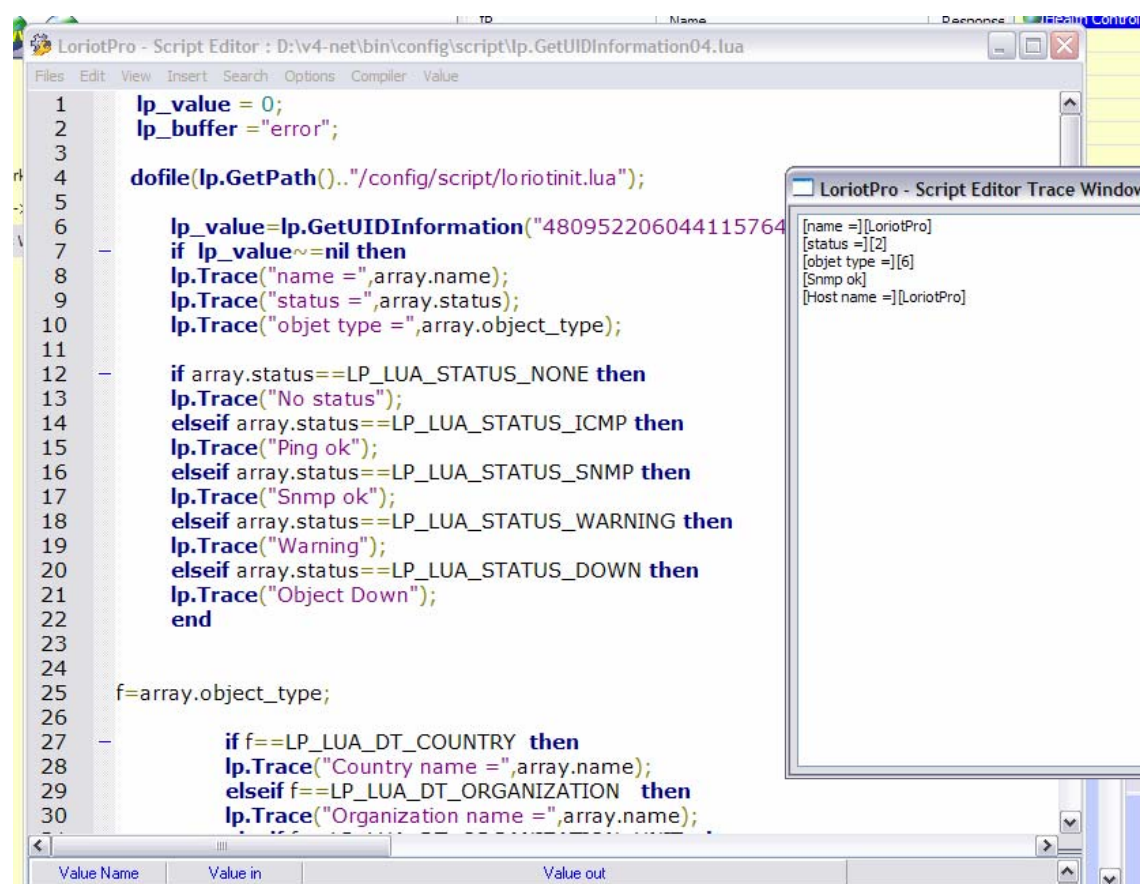
```

```
if array.status==LP_LUA_STATUS_NONE then
lp.Trace("No status");
elseif array.status==LP_LUA_STATUS_ICMP then
lp.Trace("Ping ok");
elseif array.status==LP_LUA_STATUS_SNMP then
lp.Trace("Snmp ok");
elseif array.status==LP_LUA_STATUS_WARNING then
lp.Trace("Warning");
elseif array.status==LP_LUA_STATUS_DOWN then
lp.Trace("Object Down");
end

f=array.object_type;

    if f==LP_LUA_DT_COUNTRY then
lp.Trace("Country name =",array.name);
    elseif f==LP_LUA_DT_ORGANIZATION then
lp.Trace("Organization name =",array.name);
    elseif f==LP_LUA_DT_ORGANIZATION_UNIT then
lp.Trace("Organization Unit name =",array.name);
    elseif f==LP_LUA_DT_NETWORK then
lp.Trace("Network name =",array.name);
    elseif f==LP_LUA_DT_HOST then
lp.Trace("Host name =",array.name);
    elseif f==LP_LUA_DT_HOST_PLUGIN then
lp.Trace("plugin name =",array.name);
    end
lp_buffer ="ok";

end
```



The screenshot displays the LoriotPro Script Editor window with a Lua script and a Trace Window. The script is as follows:

```
1  lp_value = 0;
2  lp_buffer = "error";
3
4  dofile(lp.GetPath().."/config/script/loriotinit.lua");
5
6  lp_value=lp.GetUIDInformation("480952206044115764");
7  if lp_value~=nil then
8    lp.Trace("name =",array.name);
9    lp.Trace("status =",array.status);
10   lp.Trace("objet type =",array.object_type);
11
12   if array.status==LP_LUA_STATUS_NONE then
13     lp.Trace("No status");
14   elseif array.status==LP_LUA_STATUS_ICMP then
15     lp.Trace("Ping ok");
16   elseif array.status==LP_LUA_STATUS_SNMP then
17     lp.Trace("Snmp ok");
18   elseif array.status==LP_LUA_STATUS_WARNING then
19     lp.Trace("Warning");
20   elseif array.status==LP_LUA_STATUS_DOWN then
21     lp.Trace("Object Down");
22   end
23
24
25  f=array.object_type;
26
27   if f==LP_LUA_DT_COUNTRY then
28     lp.Trace("Country name =",array.name);
29   elseif f==LP_LUA_DT_ORGANIZATION then
30     lp.Trace("Organization name =",array.name);
```

The Trace Window on the right shows the following output:

```
[name =][LoriotPro]
[status =][2]
[objet type =][6]
[Snmp ok]
[Host name =][LoriotPro]
```

At the bottom of the editor, there is a table with the following headers:

Value Name	Value in	Value out
------------	----------	-----------

Ip.GetChildUID

```
childnumber=Ip.GetChildUID(uid,"arrayname");
```

Explication

childnumber=Ip.GetChildUID(uid,"arrayname"); permet de récupérer sous la forme d'un tableau les UID des enfants d'un objet de la directory. Cette fonction permet de « browser » la directory à la recherche d'une information.

Paramètres en entrée

« uid » Une valeur sur 64Bits sous la forme d'une chaîne de caractère qui est le référentiel d'un objet de la directory « le père » .
 « arrayname » Un nom sous la forme d'une chaîne de caractère pour le tableau de stockage des réponses.

Paramètres en sortie

Result =1 = si des enfants ou nil si une erreur.

Si Result =1 un tableau

V400 b138 SP0-bx 25 mars 2006 :

```
modify : LUA Ip.GetChildUID('uid','array'); if uid egal 'word' this array return all directory objects
        Ip.GetChildUID('word','array');
```

Variables en retour	description
arrayname[number]	Le nombre d'objet enfant du père sous la forme d'une valeur.
arrayname[UIDx]	UID 64bits associé aux objets enfants x sous la forme d'une chaîne de caractères. Si array.number = 2 arrayname['UID0'] = UDI premier enfant arrayname['UID1'] = UDI deuxieme enfant

Exemple

```
Ip_value = 0;
Ip_buffer ="error";

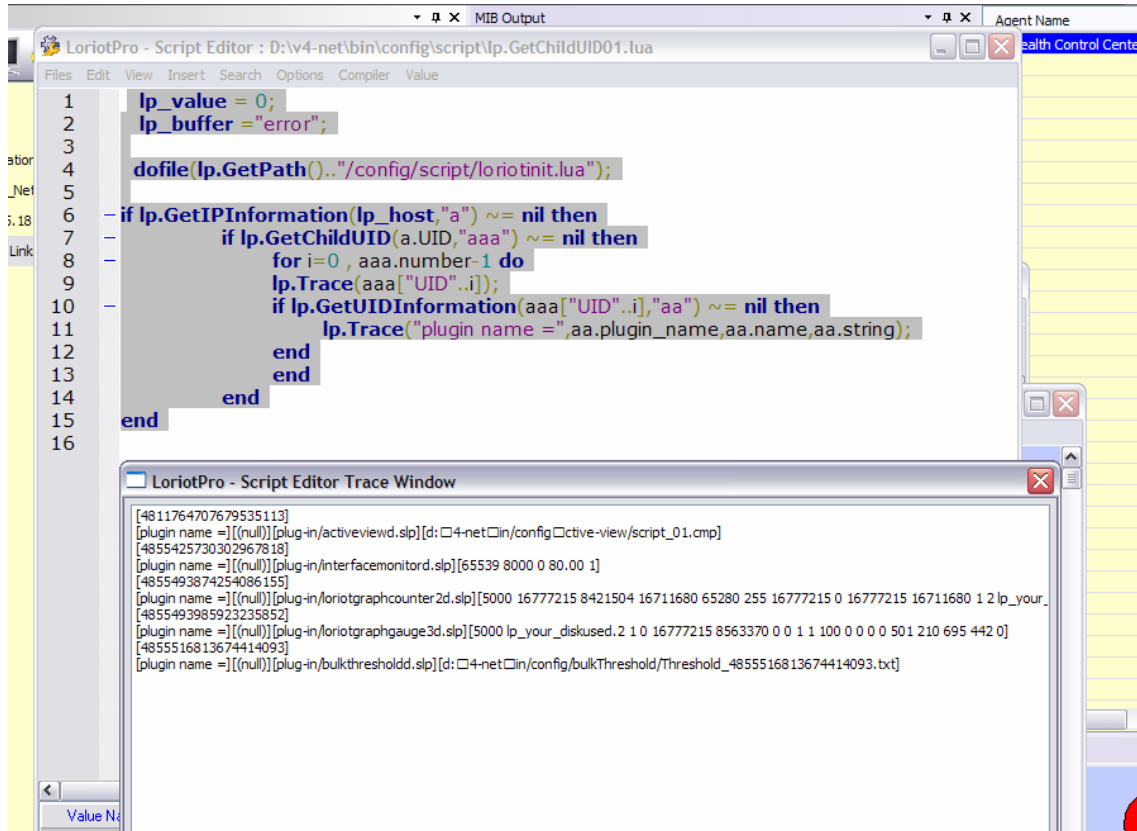
dofile(Ip.GetPath().."/config/script/loriotinit.lua");

if Ip.GetIPInformation(Ip_host,"a") ~= nil then
    if Ip.GetChildUID(a.UID,"aaa") ~= nil then
```

```

        for i=0 , aaa.number-1 do
        lp.Trace(aaa["UID"..i]);
        if lp.GetUIDInformation(aaa["UID"..i],"aa") ~= nil then
            lp.Trace("plugin name =",aa.plugin_name,aa.name,aa.string);
        end
        end
    end
end
end
end

```



Ip.atoi

```
value=Ip.atoi("ip");
```

Explication

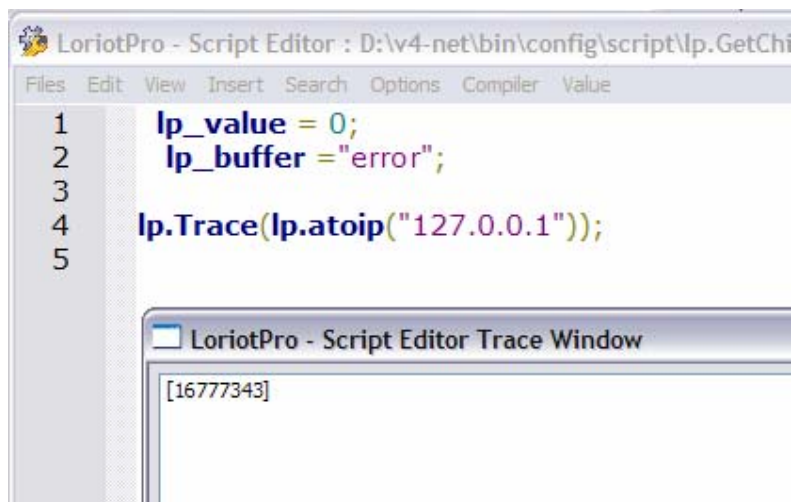
value=Ip.atoi("ip"); convertie une adresse ip de type chaîne de caractères vers un type long.

Paramètres en entrée

« ip » Une adresse IP sous la forme d'une chaîne de caractère (« xxx.xxx.xxx.xxx »).

Paramètres en sortie

Un long représentant l'adresse IP ou nil



Ip.iptoa

```
string=ip.iptoa(long);
```

Explication

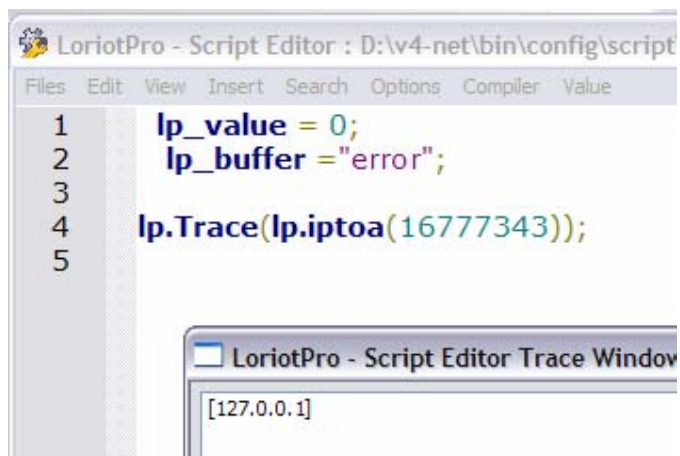
string=ip.iptoa(long); convertie une adresse ip de type long vers une chaîne de caractères.

Paramètres en entrée

long une adresse ip sous la forme d'un long (32bits)

Paramètres en sortie

Une adresse IP sous la forme d'une chaîne de caractères (« xxx.xxx.xxx.xxx ») ou nil.



Ip.HexToOID

```
‘.oid-string’=Ip.HexToOID(‘xx:xx:xx:xx:xx:xx:xx:xx:’);
```

Explication

HexToOID convertie une chaine de caractère hexadecimal en.

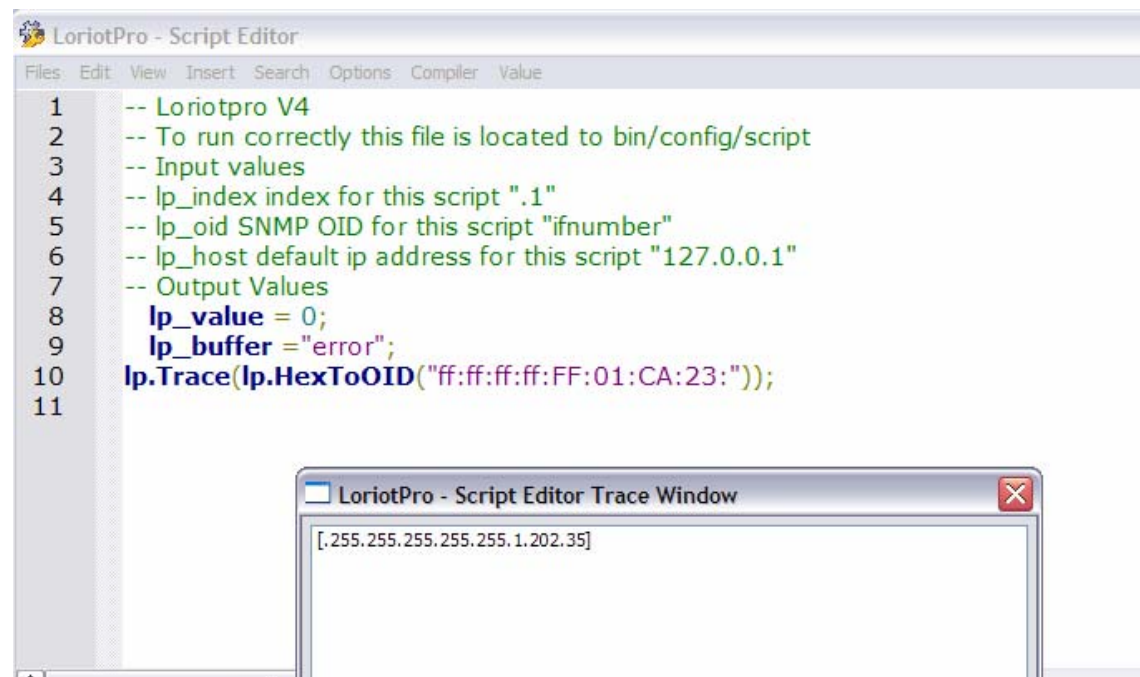
Paramètres en entrée

‘xx:xx:xx:xx:xx:xx:xx:xx:’ une chaîne de caractères sous la forme xy:xy:

Paramètres en sortie

Une chaine de caractères sous la forme d’un OID .xxx.xxx.xxx ou nil.

Exemple



The screenshot shows the 'LoriotPro - Script Editor' window with a menu bar (Files, Edit, View, Insert, Search, Options, Compiler, Value) and a code editor containing the following Lua script:

```
1 -- Loriotpro V4
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 Ip.Trace(Ip.HexToOID("ff:ff:ff:ff:FF:01:CA:23:"));
11
```

Below the script editor, a 'LoriotPro - Script Editor Trace Window' is open, displaying the output of the script: [255.255.255.255.1.202.35].

Ip.GetIPFromMAC

```
“ip”=Ip.GetIPFromMAC("mac_adresse");
```

Explication

“ip”=Ip.GetIPFromMAC("xx :xx :xx :xx :xx :xx"); retourne l’adresse IP du host possédant l’adresse MAC. LoriotPro recherche dans la directory la valeur et ne fait aucune requête. Si l’adresse MAC n’est pas dans la directory LoriotPro retourne nil.

Paramètres en entrée

« ip » Une adresse MAC sous la forme d’une chaîne de caractère (aa:ee:ee:ee:ee:10).

Paramètres en sortie

L’adresse IP trouvé dans la directory ou nil

Ip.FindName

```
“uid”=Ip.FindName("name");
```

Explication

Cette fonction retourne l’uid du premier objet de la directory contenant le « name » (exact) ou retourne nil.

Paramètres en entrée

« name » une chaine de caractères contenant le nom exact (no case sensitive) recherché dans la directory.

Paramètres en sortie

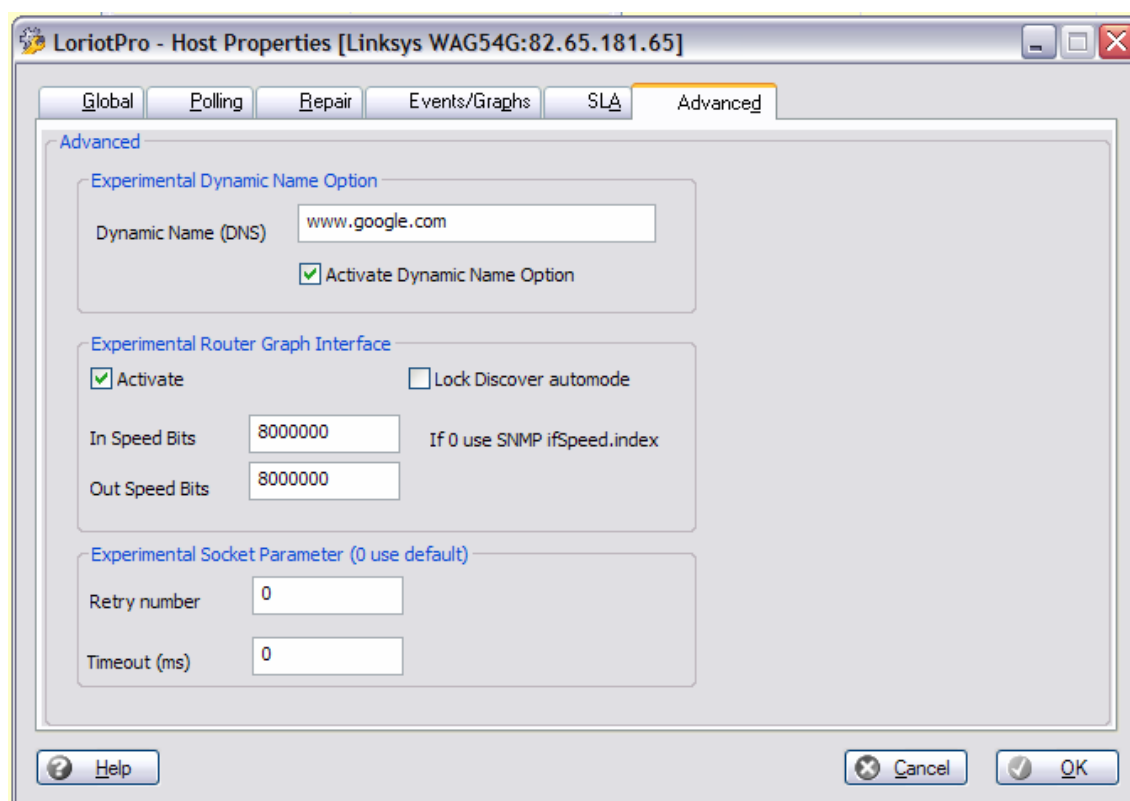
L’ « UID » de l’objet trouvé dans la directory ou nil

Ip.FindIPFronDynamicDNS

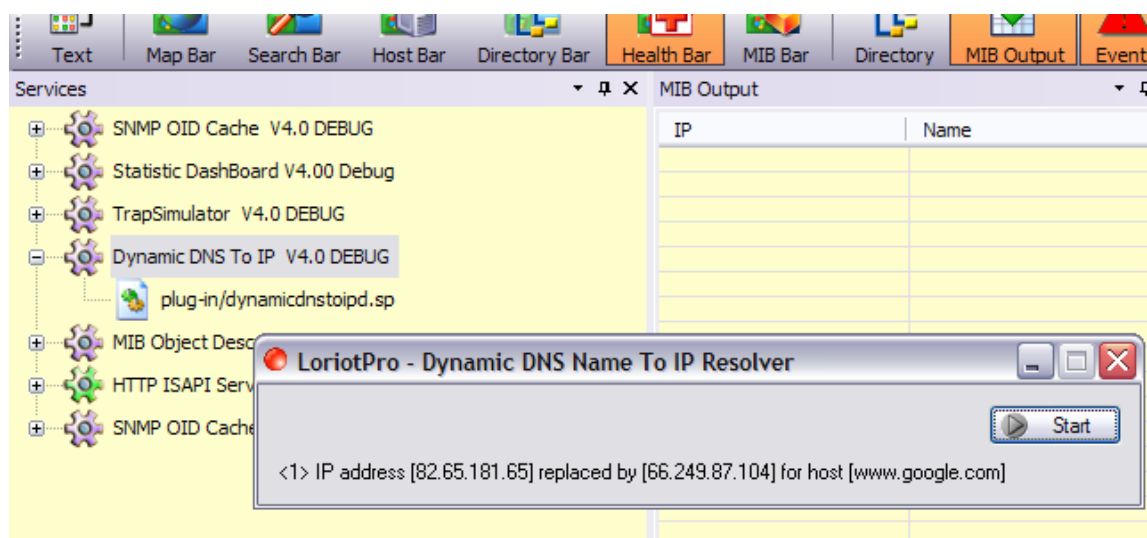
“ip”=Ip.FindIPFronDynamicDNS ("name");

Explication

“ip”=Ip.FindIPFronDynamicDNS ("name"); retourne l’adresse IP du host possédant le nom dynamique « name ». C’est une option associée e à un host de la directory qui peut avoir une adresse « mouvante » associée à un nom DNS.



Cette option utilise un plugin de Service qui à intervalle régulier va mettre à jour l’adresse IP du Host.

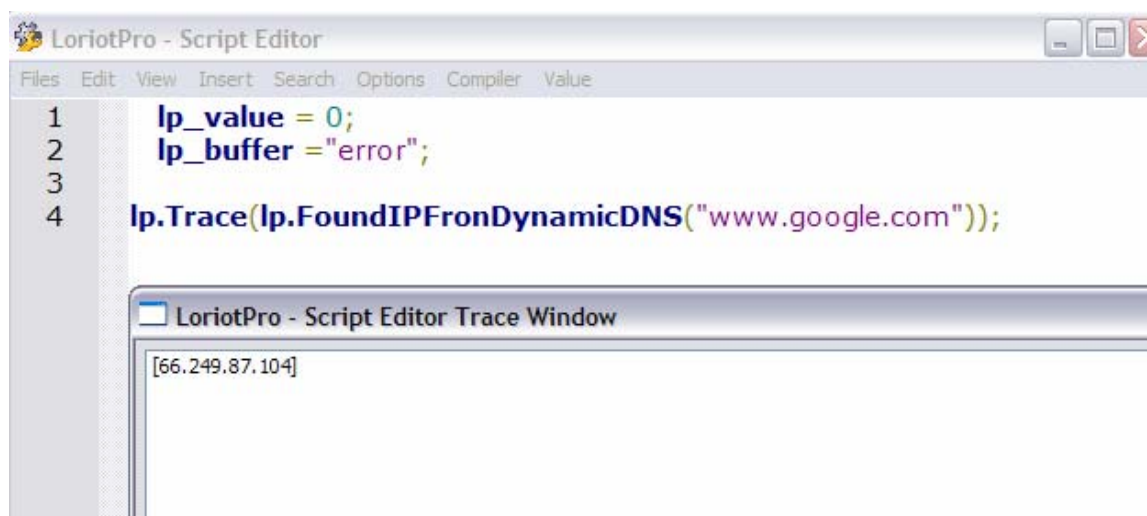


Paramètres en entrée

« name » le nom DNS d'un host de la directory configuré avec ce nom.

Paramètres en sortie

L'adresse IP trouvé dans la directory ou nil



Ip.SendEvent

```
result=Ip.SendEvent(eventnumber,level,'ipref','ipmask','buffer');
```

Explication

result=Ip.SendEvent(eventnumber,level,'ipref','ipmask','buffer'); permet d'envoyer une alarme au module de gestion d'événement de LoriotPro.

Paramètres en entrée

Eventnumber le numéro d'Event désiré.
 Attention votre numéro d'Event doit être enregistré pour que LoriotPro réalise une analyse de vos alarmes. Voir la documentation de LoriotPro.

Level le level (niveau de sécurité) associé a l'Event de 0 a 10
« ipref » l'adresse ip de référence ou un wilcard (*) (0.0.0.0)
« ipmask » le masque associé à l'adresse IP ou au réseau (10.0.0.0 255.0.0.0)
« buffer » le texte associé à l'alarme.

Paramètres en sortie

Result =1 si l'alarme est générée ou nil

Exemple

```
Ip_value = 0;  
Ip_buffer ="error";  
Ip.SendEvent( 1000,5,"127.0.0.1","255.255.255.255","test send event");
```

LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

The screenshot displays the 'LoriotPro - Script Editor' window. The main text area contains the following Lua code:

```
1  lp_value = 0;  
2  lp_buffer = "error";  
3  
4  result=lp.SendEvent(1000,5,"127.0.0.1","255.255.255.255", "event test from LUA");
```

Below the code editor is a table with the following columns: Value Name, Value in, and Value out.

Value Name	Value in	Value out
lp_host	66.249.87.99	66.249.87.99
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		error

At the bottom of the window is an 'Events' log table with columns: TimeStamp, LoriotPro A..., Ref N°, IP Ref, and Alert.

TimeStamp	LoriotPro A...	Ref N°	IP Ref	Alert
Sat Nov 05 15:59:49 ...	Local	1000	[LoriotPr...	event test from lua
Sat Nov 05 15:59:47 ...	Local	1000	[LoriotPr...	event test from lua
Sat Nov 05 15:59:36 ...	Local	1000	[LoriotPr...	event test from lua
Sat Nov 05 15:58:14 ...	Local	401	[Linksys ...	<1> ip address [66.249.87.99] not replaced by [66.249.87.104] for host [www.gc
Sat Nov 05 15:57:14 ...	Local	401	[Linksys ...	<1> ip address [66.249.87.99] not replaced by [66.249.87.104] for host [www.gc
Sat Nov 05 15:55:14 ...	Local	401	[Linksys ...	<1> ip address [66.249.87.99] not replaced by [66.249.87.104] for host [www.gc

Ip.SendExternEvent

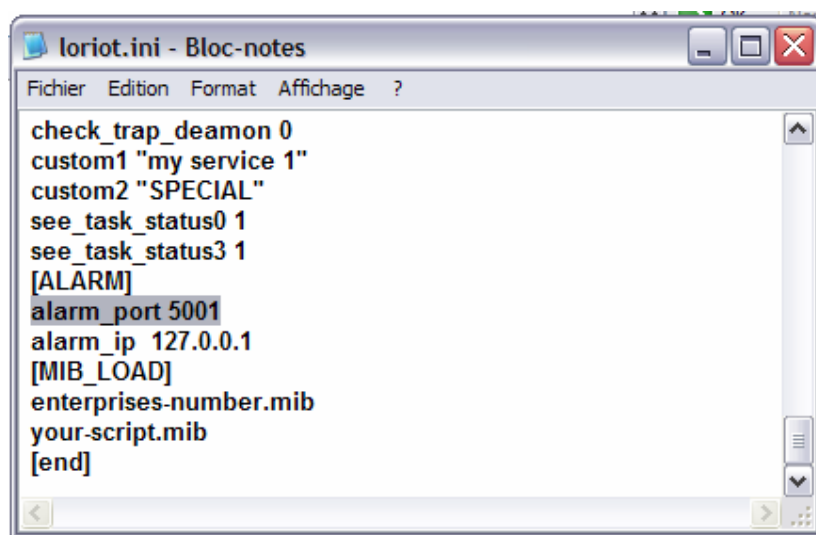
```
result=Ip.SendExternEvent('ipdest',port,eventnumber,level,'ipref','ipmask','buffer');
```

Explication

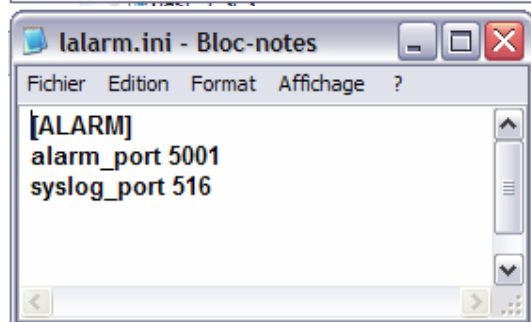
result=Ip.SendExternEvent ('ipdest',port,eventnumber,level,'ipref','ipmask','buffer');
permet d'envoyer une alarme au module d'Event d'un LoriotPro distant.

Paramètres en entrée

« ipdest » l'adresse IP du LoriotPro distant
port le numero de port UDP utilisé par le LoriotPro distant. La configuration de ce port ce réalise avec le fichier loriot.ini et lalarm.ini



```
loriot.ini - Bloc-notes
Fichier Edition Format Affichage ?
check_trap_daemon 0
custom1 "my service 1"
custom2 "SPECIAL"
see_task_status0 1
see_task_status3 1
[ALARM]
alarm_port 5001
alarm_ip 127.0.0.1
[MIB_LOAD]
enterprises-number.mib
your-script.mib
[end]
```



```
lalarm.ini - Bloc-notes
Fichier Edition Format Affichage ?
[ALARM]
alarm_port 5001
syslog_port 516
```

Eventnumber le numéro d'Event désiré.
Attention votre numéro d'Event doit être enregistré pour que LoriotPro réalise une analyse de vos alarmes. Voir la documentation de LoriotPro.

Level le level associé a l'Event de 0 a 10

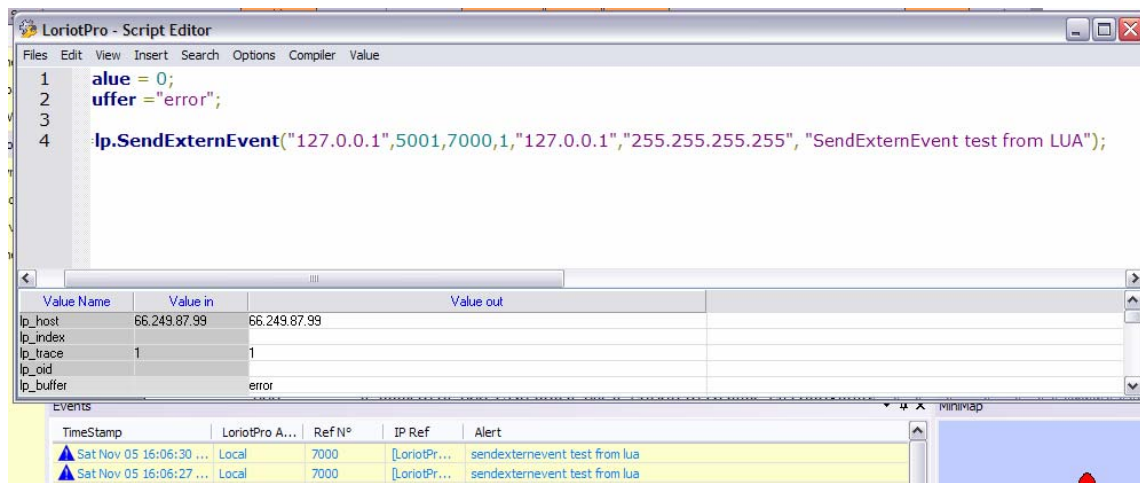
- « ipref » l'adresse ip de référence ou un wilcard (0.0.0.0)
- « ipmask » le masque associé à l'adresse IP ou au réseau (10.0.0.0 255.0.0.0)
- « buffer » le texte associé a l'alarme.

Paramètres en sortie

Result =1 si l'alarme est générée ou nil

Exemple

```
ip_value = 0;  
ip_buffer ="error";  
ip.SendExternEvent( "127.0.0.1",5001,7000,2,"127.0.0.1","255.255.255.255", "test send event");
```



Ip.IsLoadedMIBRef

```
result=Ip.IsLoadedMIBRef("description-mib");
```

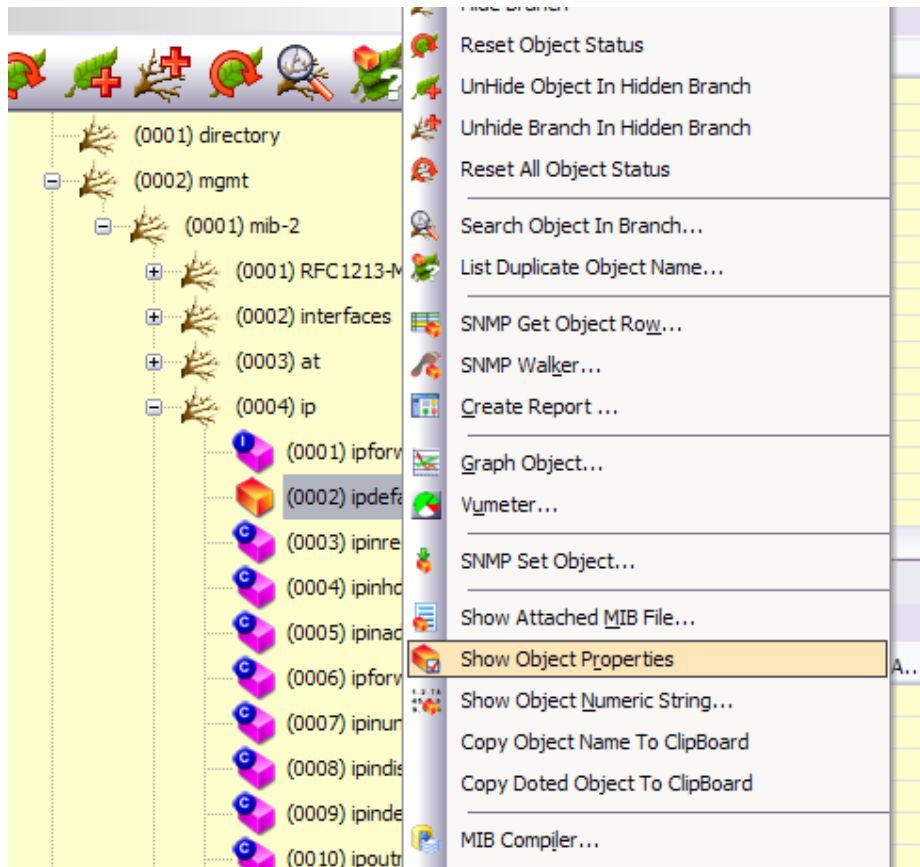
Explication

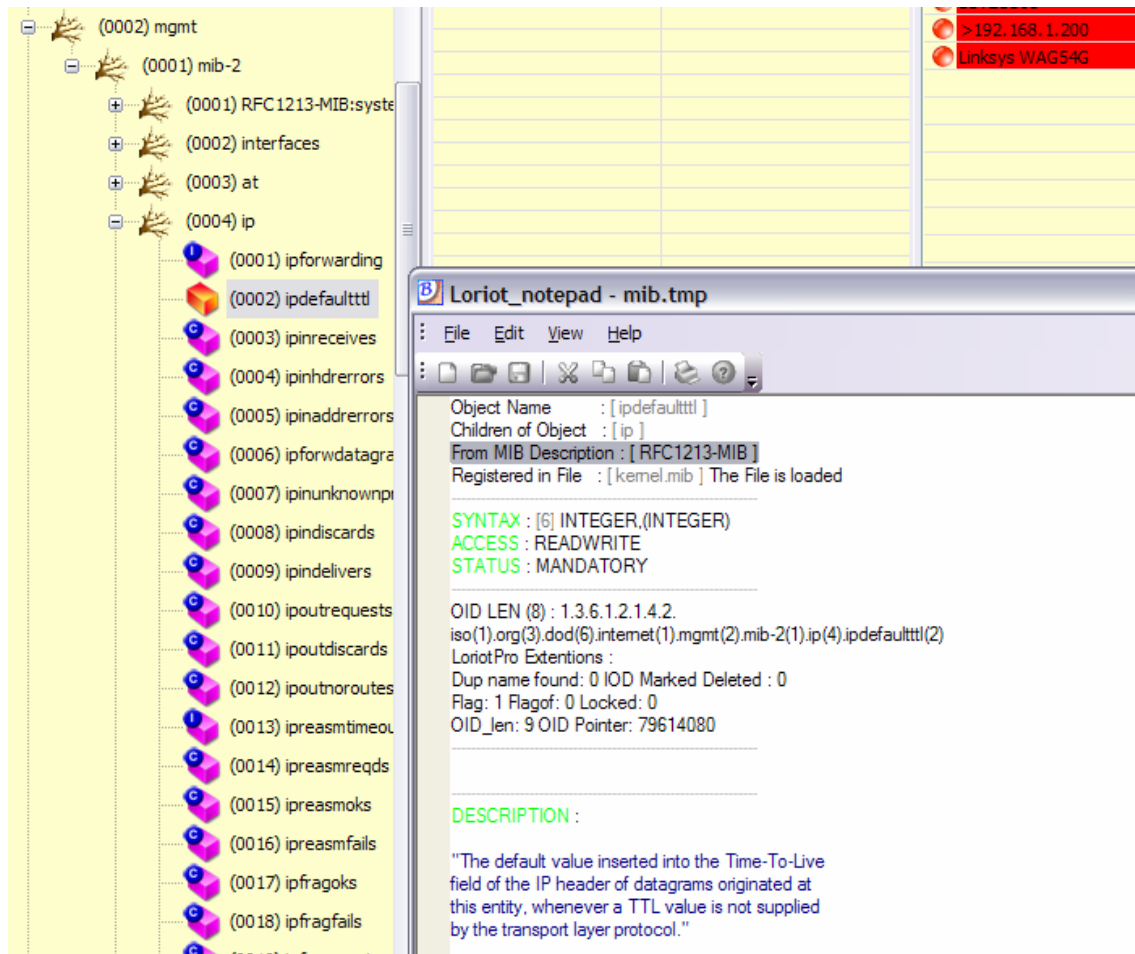
result=Ip.IsLoadedMIBRef("definitions-mib"); retourne 1 si la mib de référence « definitions-mib » est chargé en mémoire.

Au début d'un fichier de MIB il y a une clef de référence pour la MIB, tous les objets de cette MIB seront référencés comme appartenant à cette MIB. Il peut être important de vérifier qu'un ensemble d'objets SNMP (OID) est chargé avant de les utiliser.

```
-- RFC1213 MIB
-- groups in MIB-II
RFC1213-MIB DEFINITIONS ::= BEGIN
system      OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces  OBJECT IDENTIFIER ::= { mib-2 2 }
at          OBJECT IDENTIFIER ::= { mib-2 3 }
ip          OBJECT IDENTIFIER ::= { mib-2 4 }
icmp       OBJECT IDENTIFIER ::= { mib-2 5 }
tcp        OBJECT IDENTIFIER ::= { mib-2 6 }
udp        OBJECT IDENTIFIER ::= { mib-2 7 }
egp        OBJECT IDENTIFIER ::= { mib-2 8 }
-- historical (some say hysterical)
-- cmot     OBJECT IDENTIFIER ::= { mib-2 9 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }
snmp       OBJECT IDENTIFIER ::= { mib-2 11 }
```

La fonction Ip.IsLoadedMIBRef ne recherche pas par nom de fichier mais par clef de référence. Le module de compilation ou les propriétés d'un objet de l'arbre de MIB vous permettent de connaître le nom de definitions-mib associé.





Paramètres en entrée

«definitions-mib» la description de la MIB recherchée

Paramètres en sortie

Result = 1 si la MIB est chargée ou nil

Ip.IsRegisteredMIBRef

```
"file"=Ip.IsRegisteredMIBRef("description-mib ");
```

Explication

"file"=Ip.IsRegisteredMIBRef("description-mib"); retourne le nom du fichier associé si la mib de référence « definitions-mib » existe et est « registered » une MIB enregistrée comme étant existante n'est pas forcément chargée dans la database. Voir la fonction Ip.IsLoadedMIBRef

Paramètres en entrée

«definitions-mib» la description de la MIB recherchée

Paramètres en sortie

Le nom de fichier si la MIB est enregistrée ou nil

Ip.FindSNMPObject

```
Result=Ip.FindSNMPObject("named_oid");
```

Explication

Result=Ip.FindSNMPObject("named_iod"); retourne 1 si le « named_oid » est chargé dans la database.

Attention les OID en syntaxe doted (« 1.3.4.3.2 ») n'est pas supporté.

Paramètres en entrée

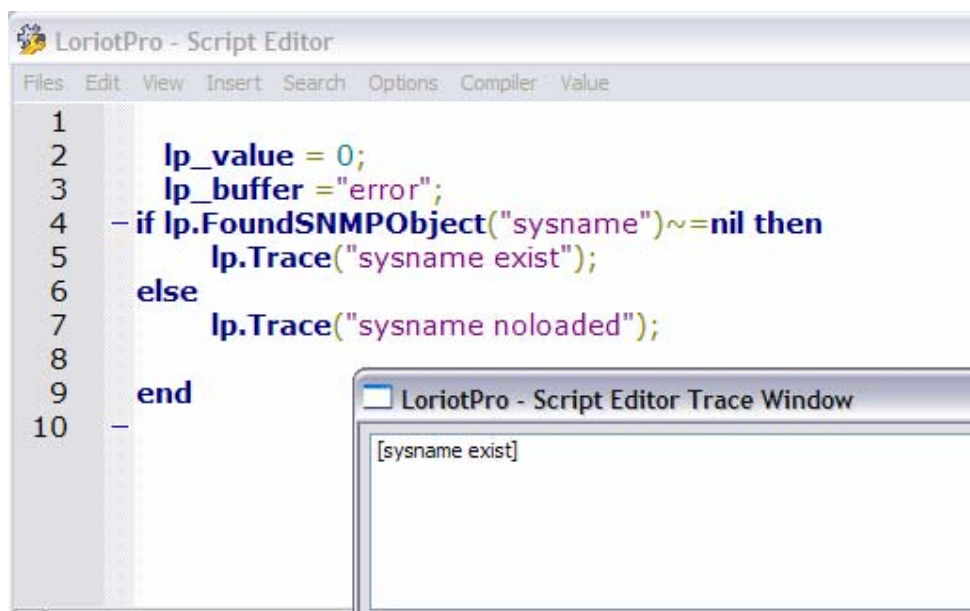
«named_oid» le nom de l'objet snmp recherché.

Paramètres en sortie

Result =1 si l'objet existe et est chargé dans la database ou nil

Exemple

```
Ip_value = 0;
Ip_buffer ="error";
if Ip.FindSNMPObject("sysname")~=nil then
    Ip.Trace("sysname exist");
else
    Ip.Trace("sysname noloaded");
end
```



Ip.GetFirstNetwork

```
“net”,“mask”=Ip.GetFirstNetwork();
```

Explication

“net”,“mask”=Ip.GetFirstNetwork(); retourne le premier network de la directory et son masque associé. Cette fonction permet d’initialiser des recherches dans la liste des network de la directory.

Attention ce n’est pas forcément le premier network visible de la directory mais le premier stocké en mémoire.

Paramètres en entrée

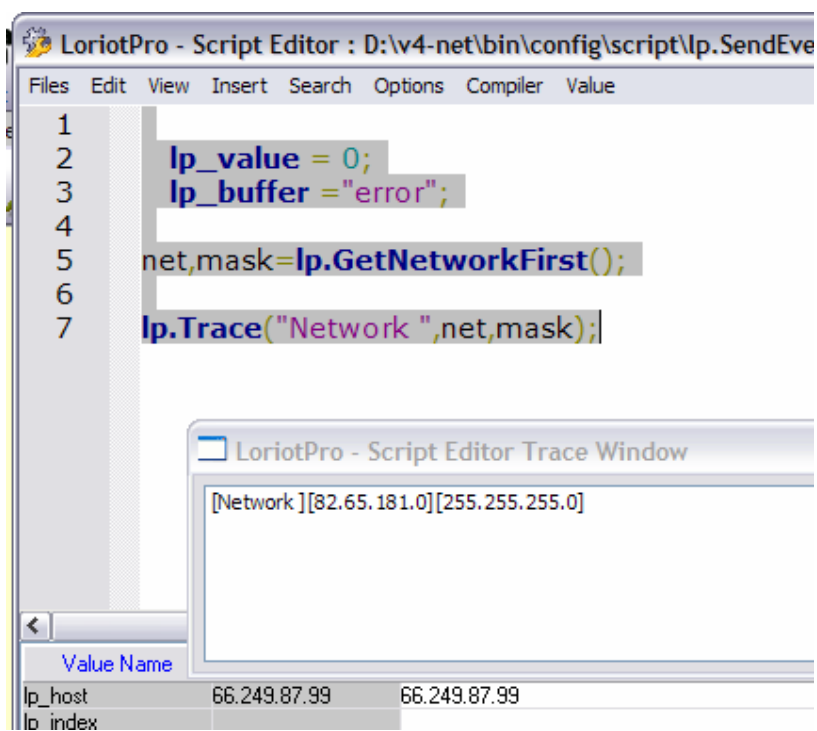
Paramètres en sortie

Nil si pas de network

« net » le network sous sa forme ip
« mask » le masque du network sous sa forme ip

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
  
net,mask=Ip.GetFirstNetwork();  
  
Ip.Trace("Network ",net,mask);
```



Ip.GetNextNetwork

```
"next_net","next_mask"=Ip.GetNextNetwork('net','mask');
```

Explication

"net","mask"=Ip.GetNextNetwork('net','mask'); retourne network suivant de la directory et son masque associé. Cette fonction permet des recherches dans la liste des networks de la directory.

Attention ce n'est pas forcément le network suivant visible de la directory mais le suivant stocké en mémoire.

Paramètres en entrée

« net » le network de départ sous sa forme ip
« mask » le masque de départ du network sous sa forme ip

Paramètres en sortie

Nil si pas de network suivant

« next_net » le network sous sa forme ip
« next_mask » le masque du network sous sa forme ip

Exemple

```
Ip_value = 0;
Ip_buffer = "error";

net,mask=Ip.GetFirstNetwork();

if net ~=nil then
a=1;
Ip.Trace(net,mask);
while a==1 do
net,mask=Ip.GetNextNetwork(net,mask);
if net==nil then a=0 end;
Ip.Trace(net,mask);
end
end
```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\lp.GetNetworkFirst01.lua

Files Edit View Insert Search Options Compiler Value

```

1
2   lp_value = 0;
3   lp_buffer = "error";
4
5   net,mask=lp.GetNetworkFirst();
6
7   -if net ~=nil then
8     a=1;
9     lp.Trace(net,mask);
10  -while a==1 do
11    net,mask=lp.GetNetworkNext(net,mask);
12    if net==nil then a=0 end;
13    lp.Trace(net,mask);
14  end
15  end
16
17

```

LoriotPro - Script Editor Trace Window

```

[82.65.181.0][255.255.255.0]
[192.168.1.0][255.255.255.0]

```

Value Name		
lp_host	66.249.87.99	66.249.87.99
lp_index		
lp_trace	1	1

Ip.GetNetworkFromIP

```
"net","mask"=Ip.GetNetworkFromIP("ip");
```

Explication

`"net","mask"=Ip.GetNetworkFromIP("ip");` retourne le network et son masque d'une adresse ip de la directory (si le network existe dans la directory), si plusieurs network sont possibles le plus approchant est retourné.

Attention ce n'est pas forcément le network (container) où se trouve le host dans la directory mais le network au sens physique.

Paramètres en entrée

« ip » l'adresse ip

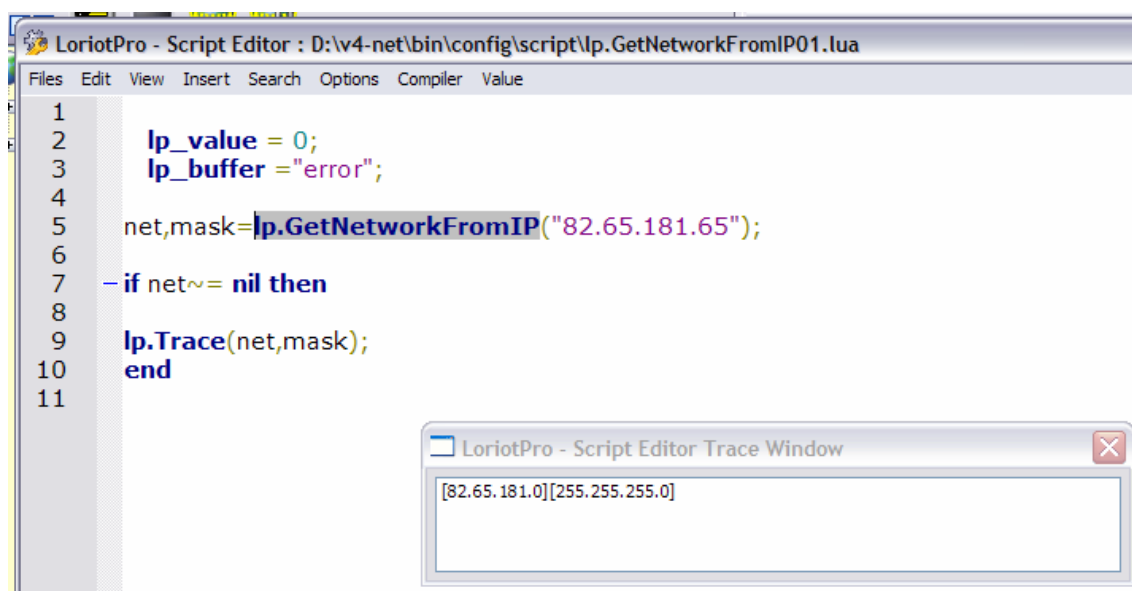
Paramètres en sortie

Nil si pas de network associé

« next_net » le network sous sa forme ip
« next_mask » le masque du network sous sa forme ip

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
  
net,mask=Ip.GetNetworkFromIP("82.65.181.65");  
  
if net~= nil then  
  
Ip.Trace(net,mask);  
end
```



The screenshot displays the LoriotPro Script Editor window with the following Lua code:

```
1  
2   lp_value = 0;  
3   lp_buffer = "error";  
4  
5   net,mask=lp.GetNetworkFromIP("82.65.181.65");  
6  
7   -if net~= nil then  
8  
9   lp.Trace(net,mask);  
10  end  
11
```

Below the script editor, a "LoriotPro - Script Editor Trace Window" is open, showing the output of the script:

```
[82.65.181.0][255.255.255.0]
```

Ip.GetNetworkInformation

```
result=Ip.GetNetworkInformation('net','mask','array');
```

Explication

result=Ip.GetNetworkInformation('net','mask','array'); retourne dans un tableau 'array' les informations spécifiques au network défini par 'net' et 'mask'.
Attention le network doit exister dans la directory.

Paramètres en entrée

« net » le network de départ sous sa forme ip
« mask » le masque de départ du network sous sa forme ip
« array » un nom pour votre tableau de réponse

Paramètres en sortie

Result=1 si 'array' est renseigné ou nil

Variables en retour	description
array['UID']	L'UID sous la forme d'une chaîne de caractères associé au réseau.
array['name']	Le nom du réseau
array['x']	La position x dans l'internetnetwork map du network
array['y']	La position y dans l'internetnetwork map du network
array['status']	Le status du network : ATTENTION : Pour calculer le status d'un network LoriotPro analyse l'ensemble des hosts appartenant au network quelles que soient leurs positions dans la directory. Le statut du container network ne correspond donc pas forcément à l'état du host le plus défavorable placé en position enfant dans la directory mais à l'état le plus défavorable de l'ensemble des hosts du network se trouvant dans la directory.
array['iftype']	Un numéro de référence du type d'interface : <pre>Loriottestentry ::= TEXTUAL-CONVENTION STATUS current DESCRIPTION "test snmpv2 ASN Mib." SYNTAX INTEGER { other(1), -- none of the following regular1822(2), hdh1822(3), ddnX25(4), rfc877x25(5), ethernetCsmacd(6), iso88023Csmacd(7), iso88024TokenBus(8), iso88025TokenRing(9), iso88026Man(10), starLan(11), proteon10Mbit(12), proteon80Mbit(13), hyperchannel(14), fddi(15),</pre>

	<pre> lapb(16), sdsl(17), dsl(18), -- DS1-MIB el(19), -- Obsolete see DS1-MIB basicISDN(20), primaryISDN(21), propPointToSerial(22), -- proprietary serial ppp(23), softwareLoopback(24), eon(25), -- CLNP over IP ethernet3Mbit(26), nslip(27), -- XNS over IP slip(28), -- generic SLIP ultra(29), -- ULTRA technologies ds3(30), -- DS3-MIB sip(31), -- SMDS, coffee frameRelay(32), -- DTE only. rs232(33), para(34), -- parallel-port arcnet(35), -- arcnet arcnetPlus(36), -- arcnet plus atm(37), -- ATM cells miox25(38), sonet(39), -- SONET or SDH x25ple(40), iso8802211c(41), localTalk(42), smdsDxi(43), frameRelayService(44), -- FRNETSERV-MIB v35(45), hssi(46), hippi(47), modem(48), -- Generic modem aal5(49), -- AAL5 over ATM sonetPath(50), sonetVT(51), smdsIcip(52), -- SMDS InterCarrier Interface propVirtual(53), -- proprietary virtual/internal propMultiplexor(54), -- proprietary multiplexing ieee80212(55), -- 100BaseVG fibreChannel(56), -- Fibre Channel hippiInterface(57), -- HIPPI interfaces frameRelayInterconnect(58), -- Obsolete use either -- frameRelay(32) or -- frameRelayService(44). aflane8023(59), -- ATM Emulated LAN for 802.3 aflane8025(60), -- ATM Emulated LAN for 802.5 cctEmul(61), -- ATM Emulated circuit fastEther(62), -- Fast Ethernet (100BaseT) isdn(63), -- ISDN and X.25 v11(64), -- CCITT V.11/X.21 v36(65), -- CCITT V.36 g703at64k(66), -- CCITT G703 at 64Kbps g703at2mb(67), -- Obsolete see DS1-MIB qlc(68), -- SNA QLLC fastEtherFX(69), -- Fast Ethernet (100BaseFX) channel(70), -- channel ieee80211(71), -- radio spread spectrum lmb370parChan(72), -- IBM System 360/370 OEMI Channel escon(73), -- IBM Enterprise Systems Connection dlsw(74), -- Data Link Switching isdns(75), -- ISDN S/T interface isdnu(76), -- ISDN U interface lapd(77), -- Link Access Protocol D ipSwitch(78), -- IP Switching Objects rsrb(79), -- Remote Source Route Bridging atmLogical(80), -- ATM Logical Port ds0(81), -- Digital Signal Level 0 ds0Bundle(82), -- group of ds0s on the same dsl bsc(83), -- Bisynchronous Protocol async(84), -- Asynchronous Protocol cnr(85), -- Combat Net Radio iso88025Dtr(86), -- ISO 802.5r DTR eplrs(87), -- Ext Pos Loc Report Sys arap(88), -- Appletalk Remote Access Protocol propCnls(89), -- Proprietary Connectionless Protocol hostPad(90), -- CCITT-ITU X.29 PAD Protocol termPad(91), -- CCITT-ITU X.3 PAD Facility frameRelayMPI(92), -- Multiproto Interconnect over FR x213(93), -- CCITT-ITU X213 adsl(94), -- Asymmetric Digital Subscriber Loop radsl(95), -- Rate-Adapt. Digital Subscriber Loop sdsl(96), -- Symmetric Digital Subscriber Loop vdsl(97), -- Very H-Speed Digital Subscrib. Loop iso88025CRFPInt(98), -- ISO 802.5 CRFP myrinet(99), -- Myricom Myrinet voiceEM(100), -- voice receive and transmit voiceFXO(101), -- voice Foreign Exchange Office voiceFXS(102), -- voice Foreign Exchange Station voiceEncap(103), -- voice encapsulation voiceOverIp(104), -- voice over IP encapsulation atmDxi(105), -- ATM DXI atmFuni(106), -- ATM FUNI atmIma(107), -- ATM IMA pppMultilinkBundle(108), -- PPP Multilink Bundle ipOverCdic(109), -- IBM ipOverCdic ipOverClaw(110), -- IBM Common Link Access to Workstn stackToStack(111), -- IBM stackToStack virtualIpAddress(112), -- IBM VIPA mpc(113), -- IBM multi-protocol channel support ipOverAtm(114), -- IBM ipOverAtm iso88025Fiber(115), -- ISO 802.5j Fiber Token Ring tdlc(116), -- IBM twinaxial data link control gigabitEthernet(117), -- Gigabit Ethernet hdlc(118), -- HDLC lapf(119), -- LAP F v37(120), -- V.37 x25mlp(121), -- Multi-Link Protocol x25huntGroup(122), -- X25 Hunt Group trasnHdlc(123), -- Transp HDLC interleave(124), -- Interleave channel fast(125), -- Fast channel ip(126), -- IP (for APFN HPR in IP networks) docsCableMacLayer(127), -- CATV Mac Layer docsCableDownstream(128), -- CATV Downstream interface docsCableUpstream(129), -- CATV Upstream interface al2MppSwitch(130), -- Avalon Parallel Processor tunnel(131), -- Encapsulation interface coffee(132), -- coffee pot ces(133), -- Circuit Emulation Service atmSubInterface(134), -- ATM Sub Interface l2vlan(135), -- Layer 2 Virtual LAN using 802.1Q l3ipvlan(136), -- Layer 3 Virtual LAN using IP </pre>
--	--

	<pre> l3ipxvlan (137), -- Layer 3 Virtual LAN using IPX digitalPowerline (138), -- IP over Power Lines mediaMailOverIp (139), -- Multimedia Mail over IP dtm (140), -- Dynamic synchronous Transfer Mode dcm (141), -- Data Communications Network ipForward (142), -- IP Forwarding Interface msdsl (143), -- Multi-rate Symmetric DSL ieee1394 (144), -- IEEE1394 High Performance Serial Bus if-gsn (145), -- HIPPI-6400 dvbRccMacLayer (146), -- DVB-RCC MAC Layer dvbRccDownstream (147), -- DVB-RCC Downstream Channel dvbRccUpstream (148), -- DVB-RCC Upstream Channel atmVirtual (149), -- ATM Virtual Interface mplsTunnel (150), -- MPLS Tunnel Virtual Interface srp (151), -- Spatial Reuse Protocol voiceOverAtm (152), -- Voice Over ATM voiceOverFrameRelay (153), -- Voice Over Frame Relay idsl (154), -- Digital Subscriber Loop over ISDN compositeLink (155), -- Avici Composite Link Interface ss7SigLink (156), -- SS7 Signaling Link propWirelessP2P (157), -- Prop. P2P wireless interface frForward (158), -- Frame Forward Interface rfcl483 (159), -- Multiprotocol over ATM AAL5 usb (160), -- USB Interface ieee8023adLag (161), -- IEEE 802.3ad Link Aggregate bgppolicyaccounting (162), -- BGP Policy Accounting frf16MfrBundle (163), -- FRF .16 Multilink Frame Relay h323Gatekeeper (164), -- H323 Gatekeeper h323Proxy (165), -- H323 Voice and Video Proxy mpls (166), -- MPLS mfSigLink (167), -- Multi-frequency signaling link hdlsl2 (168), -- High Bit-Rate DSL - 2nd generation shdsl (169), -- Multirate HDSL2 ds1FDL (170), -- Facility Data Link 4Kbps on a DS1 pos (171), -- Packet over SONET/SDH Interface dvbAsiIn (172), -- DVB-ASI Input dvbAsiOut (173), -- DVB-ASI Output plc (174), -- Power Line Communications nfas (175), -- Non Facility Associated Signaling tr008 (176), -- TR008 gr303RDT (177), -- Remote Digital Terminal gr303IDT (178), -- Integrated Digital Terminal isup (179), -- ISUP propDocsWirelessMaclayer (180), -- prop/Maclayer propDocsWirelessDownstream (181), -- prop/Downstream propDocsWirelessUpstream (182), -- prop/Upstream hiperlan2 (183), -- HIPERLAN Type 2 Radio Interface propBWAmp2Mp (184), -- PropBroadbandWirelessAccesspt2multipt sonetOverheadChannel (185), -- SONET Overhead Channel digitalWrapperOverheadChannel (186), -- Digital Wrapper aal2 (187), -- ATM adaptation layer 2 radioMAC (188), -- MAC layer over radio links atmRadio (189) -- ATM over radio links) -- IDENTIFIER := (mib-2 11 </pre>
array['ifdescr']	Une chaîne de caractère avec la définition du network
array['speed']	Une valeur avec la vitesse en bits/seconde du média
array['nb_host']	Une valeur avec le nombre de host appartenant au network dans la directory.
array['nb_router']	Une valeur avec le nombre de routeur ayant une interface appartenant au network dans la directory.

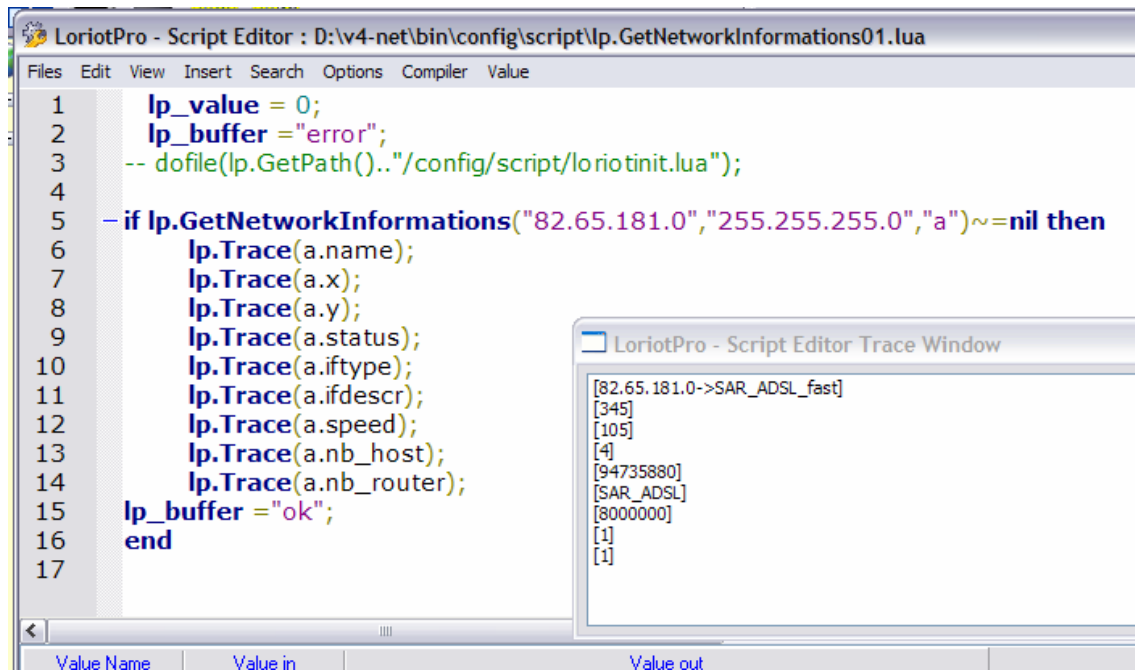
Exemple

```

lp_value = 0;
lp_buffer = "error";
-- dofile(lp.GetPath().."/config/script/loriotinit.lua");

if lp.GetNetworkInformation("82.65.181.0","255.255.255.0","a")~=nil then
    lp.Trace(a.name);
    lp.Trace(a.x);
    lp.Trace(a.y);
    lp.Trace(a.status);
    lp.Trace(a.iftype);
    lp.Trace(a.ifdescr);
    lp.Trace(a.speed);
    lp.Trace(a.nb_host);
    lp.Trace(a.nb_router);
lp_buffer = "ok";
end

```



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

```
1  lp_value = 0;
2  lp_buffer = "error";
3  -- dofile(lp.GetPath().."/config/script/loriotinit.lua");
4
5  --if lp.GetNetworkInformations("82.65.181.0","255.255.255.0","a")~=nil then
6      lp.Trace(a.name);
7      lp.Trace(a.x);
8      lp.Trace(a.y);
9      lp.Trace(a.status);
10     lp.Trace(a.iftype);
11     lp.Trace(a.ifdescr);
12     lp.Trace(a.speed);
13     lp.Trace(a.nb_host);
14     lp.Trace(a.nb_router);
15     lp_buffer = "ok";
16 end
17
```

An inset window titled "LoriotPro - Script Editor Trace Window" shows the output of the script's trace function:

```
[82.65.181.0->SAR_ADSL_fast]
[345]
[105]
[4]
[94735880]
[SAR_ADSL]
[8000000]
[1]
[1]
```

At the bottom of the editor, a table structure is visible with columns labeled "Value Name", "Value in", and "Value out".

Ip.GetFirstRouter

```
“router_id”=Ip.GetFirstRouter();
```

Explication

“ router_id ”=Ip.GetFirstRouter(); retourne la référence du premier router de la directory. Cette fonction permet d’initialiser des recherches dans la liste des routeurs de la directory.

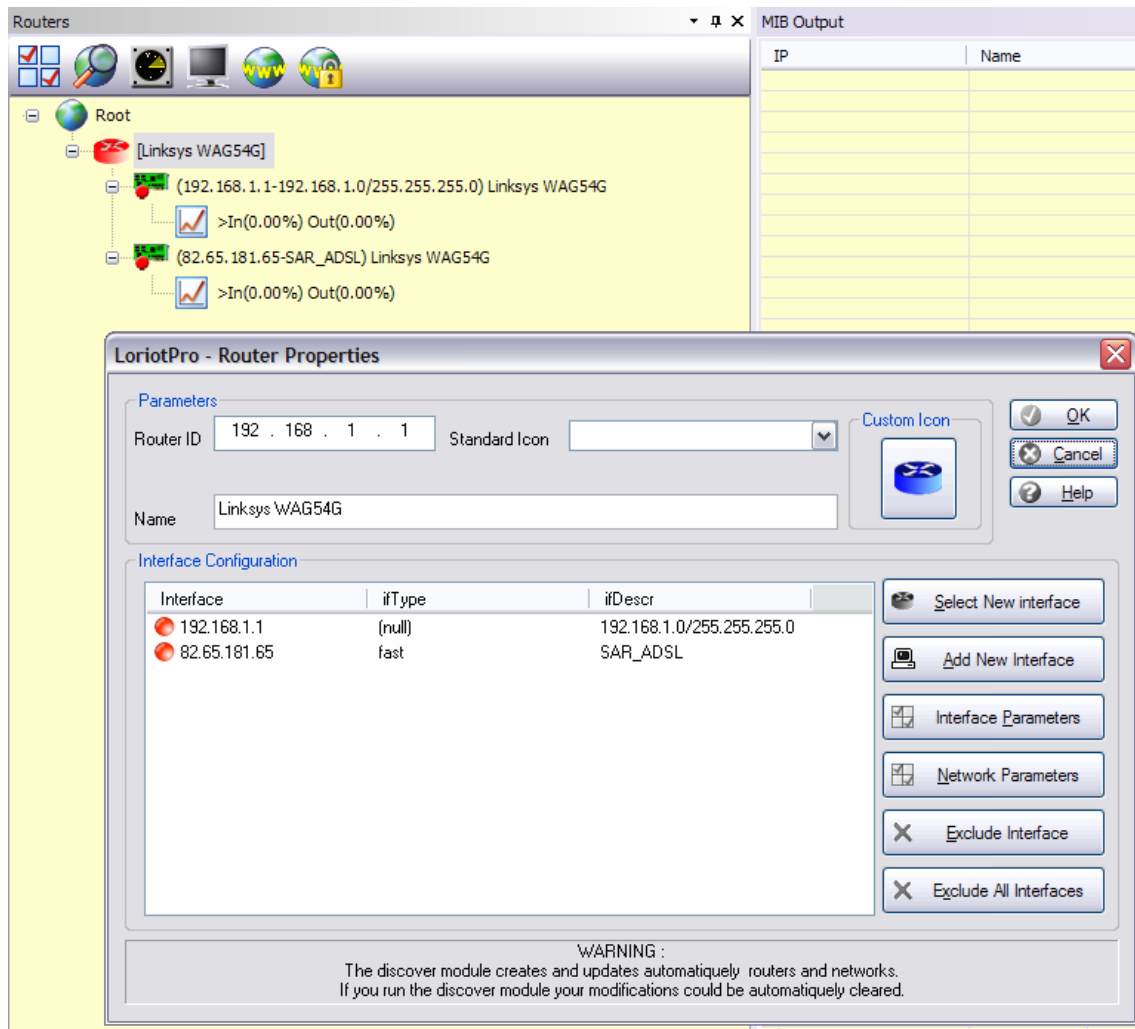
Attention ce n’est pas forcément le premier routeur visible de la directory mais le premier stocké en mémoire.

Paramètres en entrée

Paramètres en sortie

Nil si pas de routeur

« router_id » **le référentiel du router sous sa forme ip**



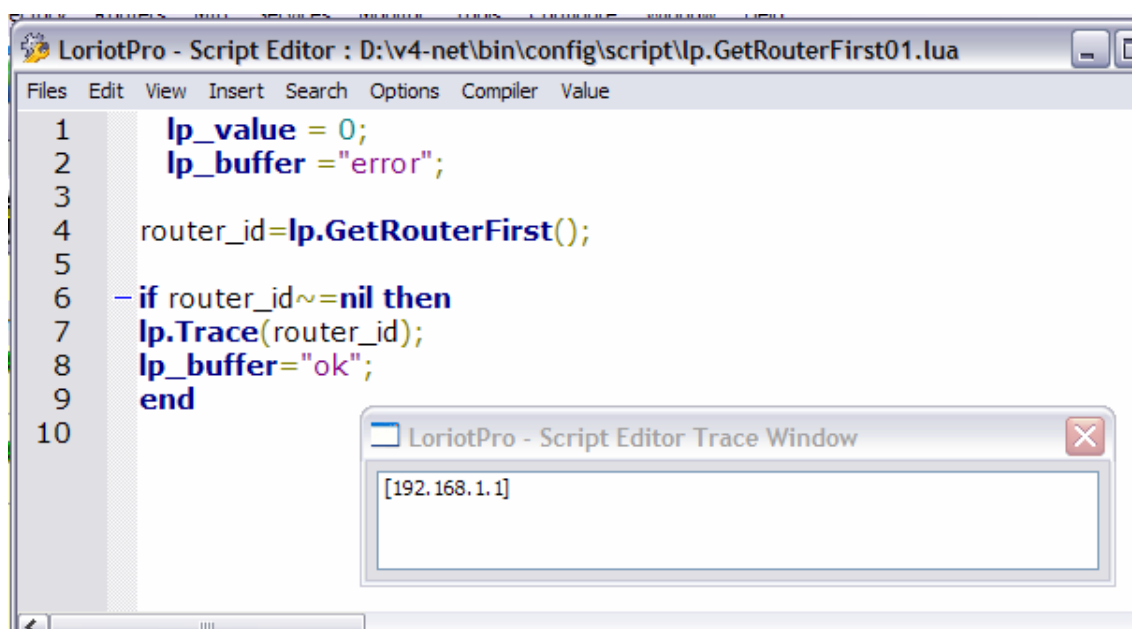
Exemple

```

ip_value = 0;
ip_buffer ="error";

router_id=lp.GetFirstRouter();

if router_id~=nil then
lp.Trace(router_id);
ip_buffer="ok";
end
    
```



Ip.GetNextRouter

```
“next_router_id”=Ip.GetNextRouter(“router_id”);
```

Explication

“next_router_id”=Ip.GetNextRouter(“router_id”); retourne le router suivant de la directory. Cette fonction permet des recherches dans la liste des routeurs de la directory.

Attention ce n’est pas forcément le routeur suivant visible de la directory mais le suivant stocké en mémoire.

Paramètres en entrée

« router_id » le router_id du routeur de départ sous sa forme ip

Paramètres en sortie

Nil si pas de routeur suivant

« next_router_id » **le router_id suivant sous sa forme ip**

Exemple

```
Ip_value = 0;
Ip_buffer = "error";

router_id=Ip.GetFirstRouter();

if router_id~=nil then
Ip.Trace(router_id);
Ip.Trace(Ip.GetNextRouter(router_id));
Ip_buffer="ok";
end
```

Ip.GetRouterInformation

```
result=Ip.GetRouterInformation(« router_id », « array ») ;
```

Explication

result=Ip.GetRouterInformation(« router_id », « array ») ; retourne dans un tableau 'array' les informations spécifique au network défini par 'net' et 'mask'.

Attention le network doit exister dans la directory.

Paramètres en entrée

« router_id » le router_id du router sous sa forme ip
« array » un nom pour votre tableau de réponse

Paramètres en sortie

Result=1 si 'array' est renseigné ou nil

Variables en retour	description
array['UID']	L'UID sous la forme d'une chaîne de caractère associé au routeur.
array ['name']	Le nom du routeur
array ['x']	La position x dans l'internetnetwork map du network
array ['y']	La position y dans l'internetnetwork map du network
array['status']	Le statut (0 – 4) du routeur
array['type']	Le type de router
array['flagused']	Le router est rattaché à un réseau

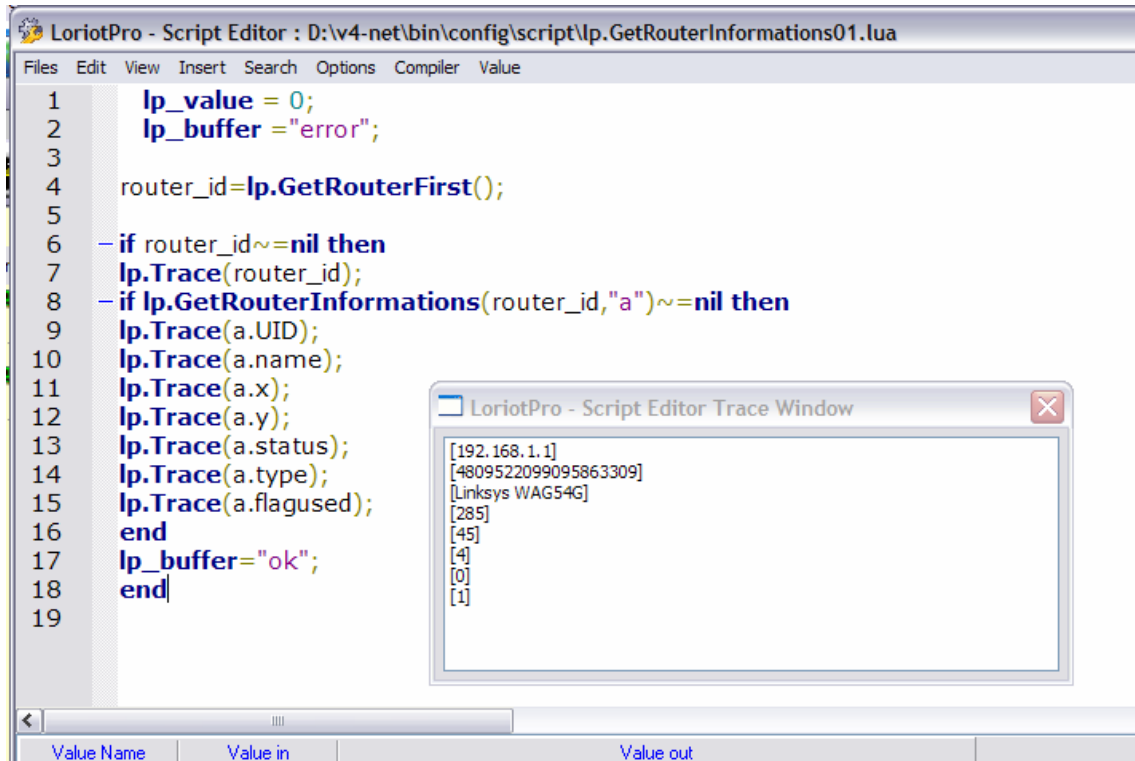
Exemple

```
Ip_value = 0;
Ip_buffer = "error";

router_id=Ip.GetFirstRouter();

if router_id~=nil then
Ip.Trace(router_id);
if Ip.GetRouterInformation(router_id,"a")~=nil then
Ip.Trace(a.UID);
Ip.Trace(a.name);
Ip.Trace(a.x);
Ip.Trace(a.y);
Ip.Trace(a.status);
Ip.Trace(a.type);
Ip.Trace(a.flagused);
```

```
end  
lp_buffer="ok";  
end
```



Ip.LoadLibrary

```
lib,init=Ip.LoadLibrary("libpath", "init_function_name"); ;
```

Explication

Cette fonction permet de charger dynamiquement dans un script un ensemble de nouvelles fonctions ou extensions du langage de script. Cette fonction implique la création d'une librairie au format LUA 5.0.2. Pour plus d'informations concernant la création de librairie d'extension consulter la documentation officiel LUA sur www.lua.org

Attention

Bien que cette fonction soit accessible à un objet SNMP virtuel, il n'est pas conseillé de l'utiliser dans ce contexte. On doit réserver cette fonction à l'extension de scripts utilisés de façon unitaire et directe à partir d'activeview par exemple.

La fermeture par l'utilisation de la fonction Ip.FreeLibrary de certaines librairies externes peut rendre le programme instable, dans ce cas ne pas utiliser cette fonction et la librairie sera fermé automatiquement à la fermeture de LoriotPro.

Paramètres en entrée

« libpath » le path et le nom de la dll a charger pour étendre les fonctions du script.
« init_function_name » le nom de la fonction d'initialisation de la librairie.

Paramètres en sortie

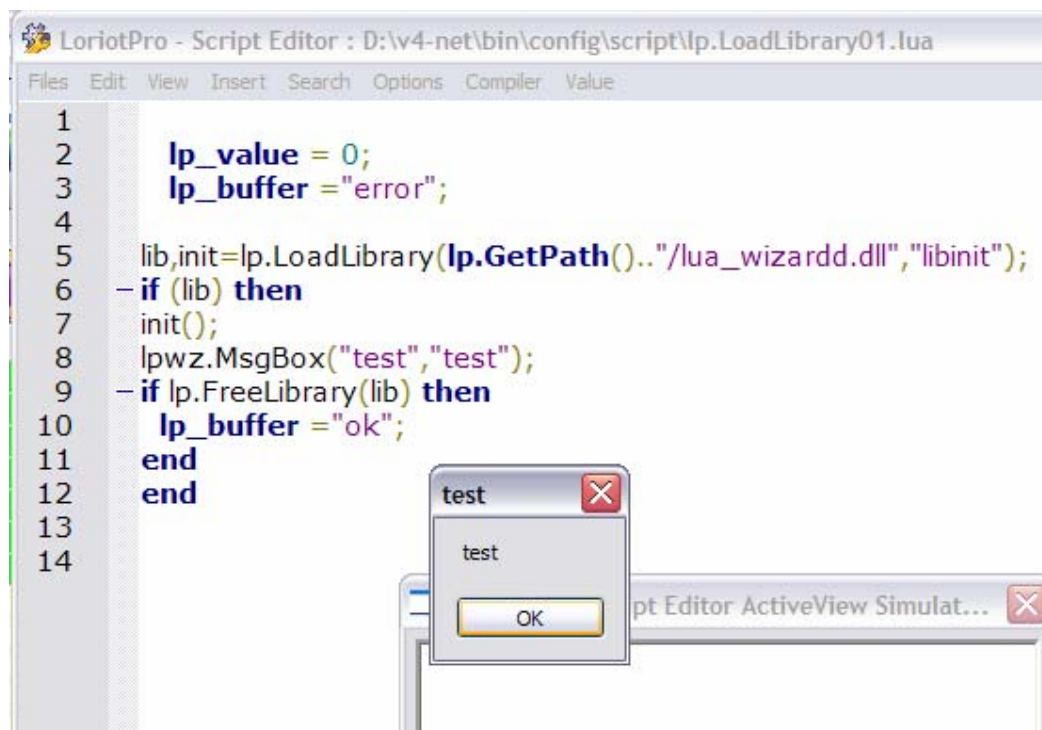
Lib un pointeur sur la librairie en mémoire pour pouvoir la libérer.
Init un pointeur sur la fonction d'initialisation de la librairie

Exemple

Cet exemple affiche un dialogue box à l'écran.

```
Ip_value = 0;
Ip_buffer = "error";

lib,init=Ip.LoadLibrary(Ip.GetPath().."/lua_wizard.dll","libinit");
if (lib) then
  init();
  Ipwz.MsgBox("test","test");
  if Ip.FreeLibrary(lib) then
    Ip_buffer = "ok";
  end
end
```



The screenshot shows the 'LoriotPro - Script Editor' window with the following Lua code:

```
1  
2     lp_value = 0;  
3     lp_buffer = "error";  
4  
5     lib,init=lp.LoadLibrary(lp.GetPath().."/lua_wizardd.dll","libinit");  
6     -if (lib) then  
7         init();  
8         lpwz.MsgBox("test","test");  
9     -if lp.FreeLibrary(lib) then  
10        lp_buffer = "ok";  
11    end  
12 end  
13  
14
```

Overlaid on the code is a 'test' dialog box with the text 'test' and an 'OK' button. Another window titled 'pt Editor ActiveView Simulat...' is partially visible in the background.

Ip.SaveDirectoryAs

```
result=Ip.SaveDirectoryAs("filename");
```

Explication

Cette fonction permet de sauvegarder la directory dans un fichier. Cette fonction force la sauvegarde des parametres liés au plugin de la directory.

```

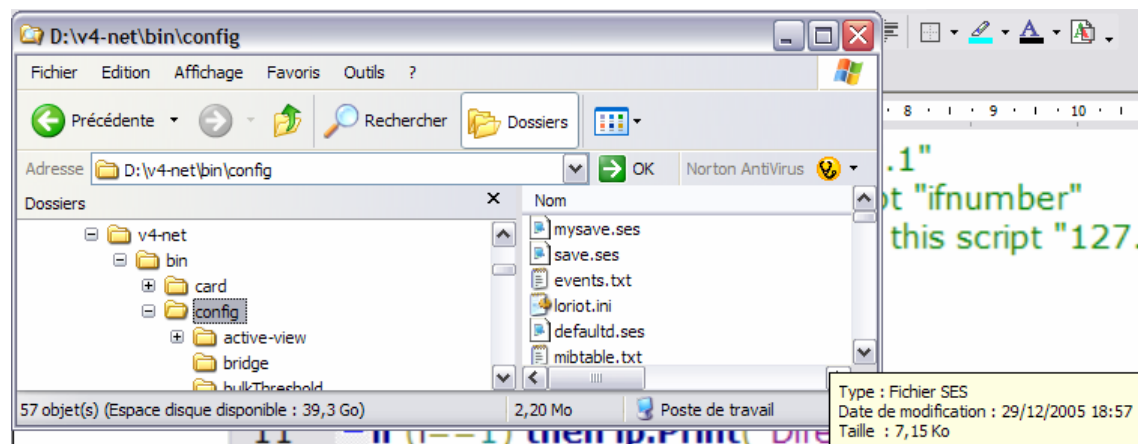
1  -- Loriotpro V4
2  -- To run correctly this file is located to bin/config/script
3  -- Input values
4  -- lp_index index for this script ".1"
5  -- lp_oid SNMP OID for this script "ifnumber"
6  -- lp_host default ip address for this script "127.0.0.1"
7  -- Output Values
8  lp_value = 0;
9  lp_buffer = "error";
10 i=Ip.SaveDirectoryAs(lp.GetPath().."/config/mysave.ses");
11 --if (i==1) then lp.Print("Directory saved")
12 else
13 lp.Print("Directory not saved")
14 end
15
16
17
    
```

Information Parsing Ended

Trace Window Value

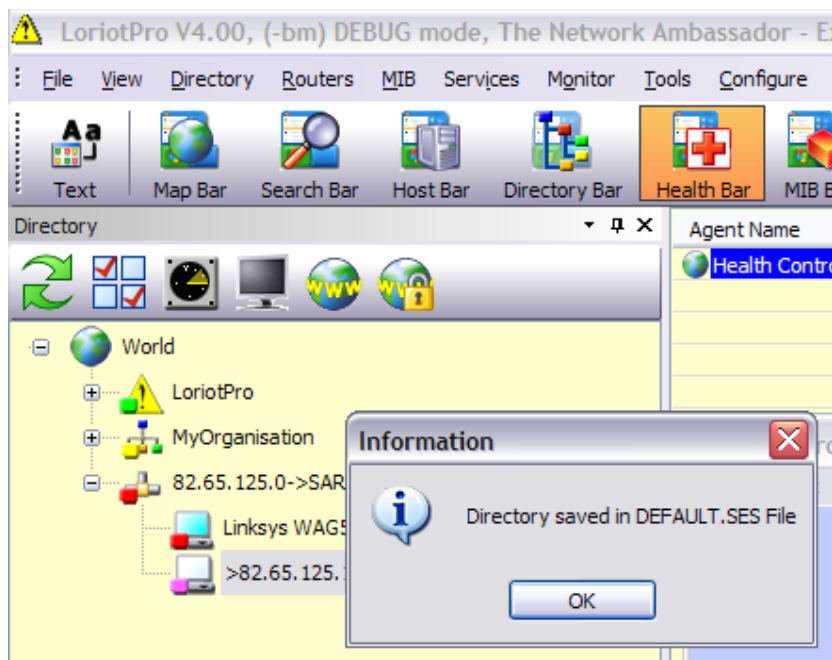
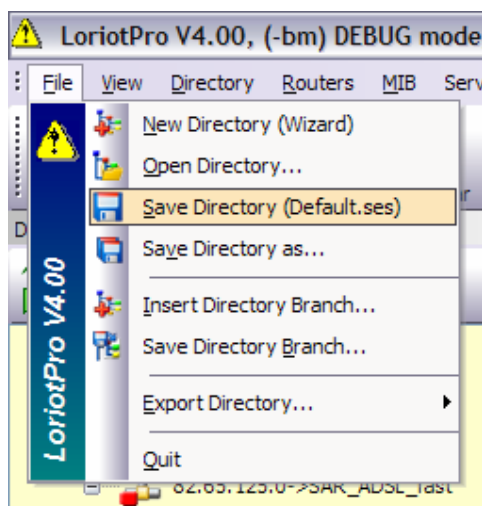
File

1 Directory saved



Attention

Il n'est pas possible de faire une sauvegarde des paramètres des plugins dans des fichiers temporaires. Seule la directory peut être sauvegardée dans des fichiers séparés mais les paramètres associés au plugin sont uniques et répartis dans l'ensemble des sous répertoires de la directory « config ». Si vous voulez faire une sauvegarde temporaire, il faut faire un save à partir du menu principal et réaliser une archive du répertoire « config/*.* » en incluant tous les sous répertoires.



Ip.FreeLibrary

```
result=Ip.FreeLibrary(lib); ;
```

Explication

Cette fonction permet de fermer une librairie d'extension préalablement ouverte par la fonction Ip.LoadLibrary.

Attention

La fermeture de certaines librairies externes peut rendre le programme instable, dans ce cas ne pas utiliser cette fonction et la librairie sera fermée automatiquement à la fermeture de loriotpro.

```
Ip.GetTableEntryList
```

```
objetnumber,indexnumber=Ip.GetTableEntryList('entry', 'array')
```

Explication

Cette fonction permet d'obtenir la liste des objets SNMP constituant une table

Paramètres en entrée

'entry' : Le nom d'un objet SNMP de type table (ifentry)
'array' : un tableau contenant les résultats de la collecte

Paramètres en sortie

objetnumber : Le nombre d'objets snmp (colonne IOD) du tableau
indexnumber : Le nombre d'index du tableau

'array' :
array[l-0] ... array[l-b] les index du tableau
array[-0] ... array[a] les noms des objets SNMP du tableau

Exemple

```
function PrintTable(entry)
a,b=Ip.GetTableEntryList(entry,"array");
if a then
for j=0,(b-1) do
Ip.Print("[",array[l-".j"],"]");
end
Ip.Print("\t");
hh="";
for j=0,(a-1) do
Ip.Print(array[j],"\t");
hh=hh..array[j];
hh=hh..";";
```

```
end
lp.Print("\n");
c,d,e=lp.GetRows(lp_host,hh,"array2");
if c then
--print line
  for j=0,(c-1) do
    --lp.Print("[",array2["I-".j],"]", "\t")
--print index
    for l=0,(e-1) do
      lp.Print("[",array2[string.format("OI-%i-%i",j,l)],"]", "\t")
    end
--print row value
    for k=0,(d-1) do
      lp.Print(array2[string.format("%s-%i",array[k],j)], "\t")
    end
    lp.Print("\n");
  end
end
end
end
```

lp.atof

value=lp.atof("string");

Explication

Cette fonction permet de forcer une chaîne de caractères en double.

Paramètres en entrée

'string' doit être un nombre sous la forme d'une chaîne de caractère sinon le résultat est 0.

Value : le valeur sous une forme double

Ip.GetAllHostFromContainer

```
number=Ip.GetAllHostFromContainer('uid','array')
```

Explication

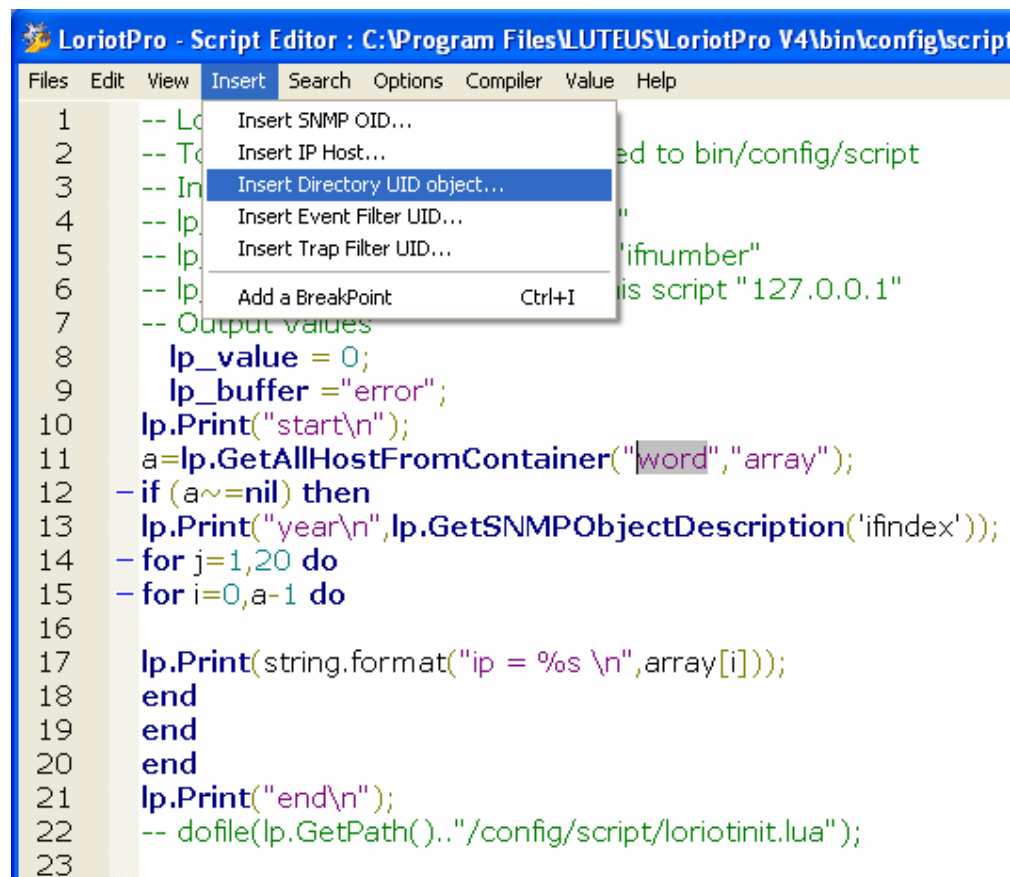
Cette fonction permet de créer une liste de host à partir de l'UID d'un élément de type container de la directory.

Paramètres en entrée

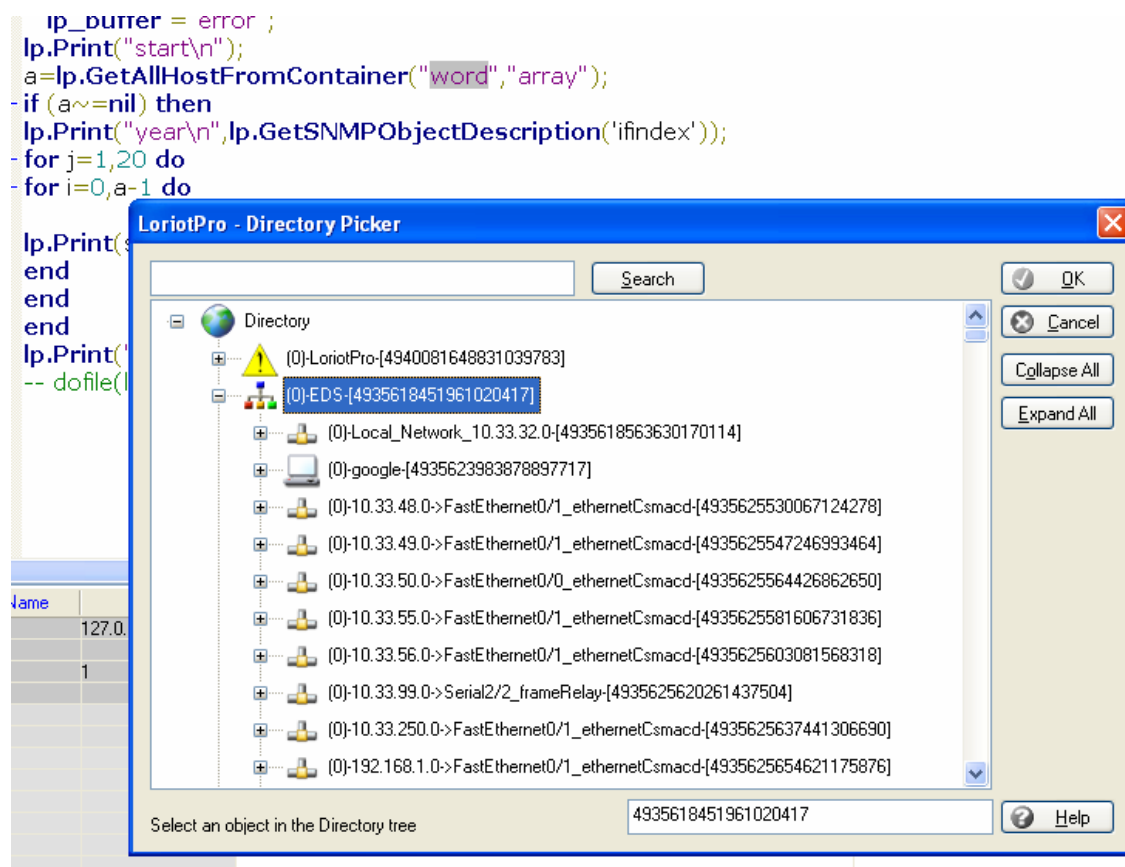
'uid' :

l'uid d'un container. L'éditeur de script contient un wizard pour vous aidez dans votre programmation.

si 'uid' est egal a « world » alors une liste complete des hosts de la directory est effectué.

The image shows a screenshot of the 'LoriotPro - Script Editor' window. The title bar indicates the file path: 'C:\Program Files\LUTEUS\LoriotPro V4\bin\config\script'. The menu bar includes 'Files', 'Edit', 'View', 'Insert', 'Search', 'Options', 'Compiler', 'Value', and 'Help'. The main editor area displays a Lua script with line numbers 1 through 23. A context menu is open over the script, listing options: 'Insert SNMP OID...', 'Insert IP Host...', 'Insert Directory UID object...' (which is highlighted), 'Insert Event Filter UID...', 'Insert Trap Filter UID...', and 'Add a BreakPoint' with a 'Ctrl+I' shortcut. The script code includes comments in green, variable assignments, and function calls like 'Ip.GetAllHostFromContainer' and 'Ip.Print'.

```
1  -- Lc
2  -- Tc
3  -- In
4  -- Ip
5  -- Ip
6  -- Ip
7  -- Output values
8  lp_value = 0;
9  lp_buffer = "error";
10 lp.Print("start\n");
11 a=Ip.GetAllHostFromContainer("word","array");
12 - if (a~=nil) then
13   lp.Print("year\n",Ip.GetSNMPObjectDescription('ifindex'));
14   - for j=1,20 do
15     - for i=0,a-1 do
16
17     lp.Print(string.format("ip = %s \n",array[i]));
18   end
19 end
20 end
21 lp.Print("end\n");
22 -- dofile(lp.GetPath().."/config/script/loriotinit.lua");
23
```



'array' : Contient les adresses IP des se trouvant dans le container de la
requete

array[0]...array[number-1]

Exemple

```

lp.Print("start\n");
a=lp.GetAllHostFromContainer("word","array");
if (a~=nil) then
for i=0,a-1 do

lp.Print(string.format("ip = %s \n",array[i]));
end
end
lp.Print("end\n");

```

Ip.GetSNMPObjectDescription

```
'string'=Ip.GetSNMPObjectDescription('oid_name')
```

Explication

Cette fonction permet de récupérer une chaîne de caractère contenant la description de l'objet snmp passé en variable.

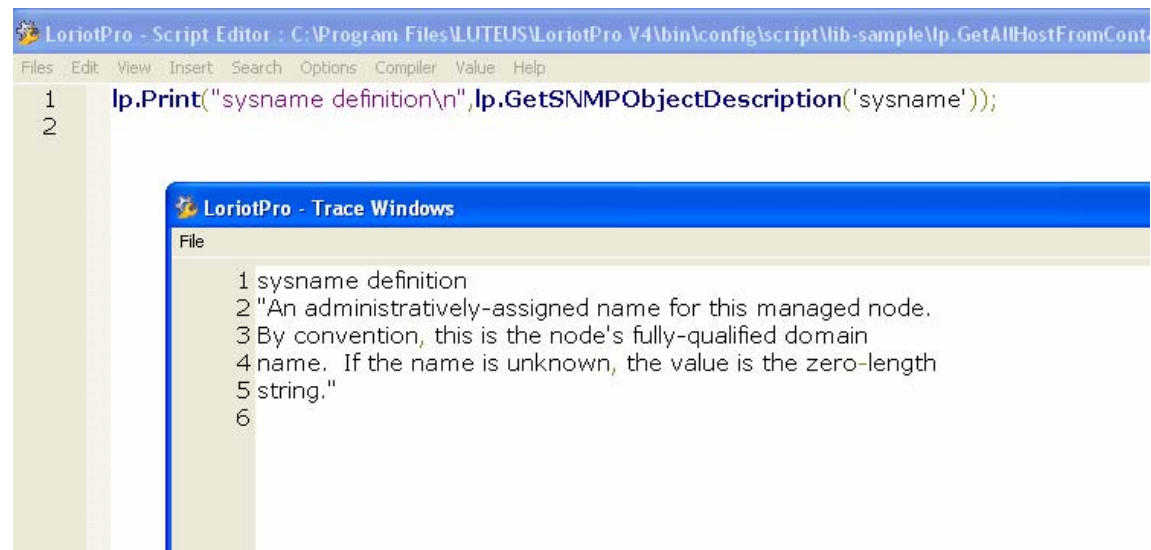
Paramètres en entrée

'oid_name' : un nom d'objet SNMP valide (non doted)

'string' :

La définition de l'objet snmp 'oid_name' si trouvé dans les fichier de MIB du répertoire *bin/mibs* .

```
Ip.Print("sysname definition\n",Ip.GetSNMPObjectDescription('sysname'));
```



Ip.GetTrapFilterList

```
number =Ip.GetTrapFilterList('array');
```

Explication

Cette fonction permet de récupérer la liste des traps connue par le logiciel et possédant des statistiques.

Paramètres en entrée

'array' : Un tableau qui contient les uid de référence des traps.

array[0]..array[number-1]

number : le nombre d'entré du tableau

Exemple

```
number=Ip.GetTrapFilterList("ta");

if number then
Ip.Print(string.format("Found %i SNMP Trap registered\n\n",number));
for j=0,(number-1) do
Ip.Print("Ack Trap -----\n");
Ip.AckTrapFilter(ta[j]);
Ip.Print(string.format("Trap uid %s\n",ta[j]));
  if Ip.GetTrapFilterInformation(ta[j],"inf") then
    Ip.Print(string.format("Trap %t%s\n",inf.trap));
    Ip.Print(string.format("\tTrap V1 Name %t%s\n",inf.name));
    Ip.Print(string.format("\tTrap V1 Generic %i Specific %i\n",inf.generic,inf.specific));
    Ip.Print(string.format("\tAssociated event number %d\n",inf.eventnumber));
    Ip.Print(string.format("\tAssociated string [%s]\n",inf.string));
    Ip.Print(string.format("\tStructure controle 0x%X\n",inf.ctrl));
    Ip.Print(string.format("\tcounter %d\n",inf.counter));
    Ip.Print(string.format("\tuid %s\n",inf.uid));

    Ip.Print(string.format("\tlast %d\n",inf.last));
    Ip.Print(string.format("\tfirst %d\n",inf.first));
    Ip.Print(string.format("\tfirst_since %d\n",inf.first_since));

    Ip.Print(string.format("\t number_total %d\n",inf.number_total));
    Ip.Print(string.format("\t number_ack %d\n",inf.number_ack));
    Ip.Print(string.format("\t number_delete %d\n",inf.number_delete));
    Ip.Print(string.format("\t number_auto_delete %d\n",inf.number_auto_delete));
    Ip.Print(string.format("\t number_since_clear %d\n",inf.number_since_clear));
    Ip.Print(string.format("\t number_not_displayed %d\n",inf.number_not_displayed));
    Ip.Print(string.format("\t number_displayed %d\n",inf.number_displayed));
    Ip.Print(string.format("\t max_display %d\n",inf.max_display));
    Ip.Print(string.format("\t number_between_action %d\n",inf.number_between_action));
    Ip.Print(string.format("\t max_action %d\n",inf.max_action));
    Ip.Print(string.format("\t max_display %d\n",inf.max_display));
    Ip.Print(string.format("\t number_between_action %d\n",inf.number_between_action));

number2=Ip.GetTrapFilterActionList(ta[j],"taa");
```



```

if number2 then
lp.Print(string.format("\nFound %i Action registered for this trap filter\n\n",number2));
for jj=0,(number2-1) do

lp.Print(string.format("\tAck Action UID [%s] for  %s %s\n",taa[jj],ta[jj],inf.trap));
lp.AckTrapFilterAction(taa[jj]);
  if lp.GetTrapFilterActionInformation(taa[jj],"infi") then
    lp.Print(string.format("\tif this parameter match : \tip \t%s mask \t %s ",infi.ip,infi.mask));
    lp.Print(string.format("\tcommunity \t%s\n",infi.community));
    lp.Print(string.format("\tSearch string value 0 [%s] condition [%i]
\n",infi.search_string0,infi.search_condition0));
    lp.Print(string.format("\tSearch string value 1 [%s] condition [%i]
\n",infi.search_string1,infi.search_condition1));
    lp.Print(string.format("\tSearch string value 2 [%s] condition [%i]
\n",infi.search_string2,infi.search_condition2));
    lp.Print(string.format("\tSearch string value 3 [%s] condition [%i]
\n",infi.search_string3,infi.search_condition3));
    lp.Print(string.format("\tSearch string value 4 [%s] condition [%i]
\n",infi.search_string4,infi.search_condition4));
    lp.Print(string.format("\tSearch string value 5 [%s] condition [%i]
\n",infi.search_string5,infi.search_condition5));
    lp.Print(string.format("\tSearch string value 6 [%s] condition [%i]
\n",infi.search_string6,infi.search_condition6));
    lp.Print(string.format("\tSearch string value 7 [%s] condition [%i]
\n",infi.search_string7,infi.search_condition7));
    lp.Print(string.format("\tSearch string value 8 [%s] condition [%i]
\n",infi.search_string8,infi.search_condition8));
    lp.Print(string.format("\tSearch string value 9 [%s] condition [%i]
\n",infi.search_string9,infi.search_condition9));

    lp.Print(string.format("\taction_type %d ",infi.action_type));

--[[
#define ACTIONTYPE_WINRUN 1
#define ACTIONTYPE_WAVE 2
#define ACTIONTYPE_SMTP 4
#define ACTIONTYPE_MAPI 8
#define ACTIONTYPE_DOSRUN 16
#define ACTIONTYPE_SYSLOG 32
#define ACTIONTYPE_TRAP 64
#define ACTIONTYPE_CUSTOM 65
#define ACTIONTYPE_SEARCH 66
#define ACTIONTYPE_NULL 67
#define ACTIONTYPE_ROUTE_EVENT 68
//v4
#define ACTIONTYPE_EVENT 69
--]]
  if inf.action_type==1 then
    lp.Print(" ACTIONTYPE_WINRUN");
  elseif inf.action_type==2 then
    lp.Print(" ACTIONTYPE_WAVE");
  elseif inf.action_type==4 then
    lp.Print(" ACTIONTYPE_SMTP");
  elseif inf.action_type==8 then
    lp.Print(" ACTIONTYPE_MAPI");
  elseif inf.action_type==16 then
    lp.Print(" ACTIONTYPE_DOSRUN");
  elseif inf.action_type==32 then
    lp.Print(" ACTIONTYPE_SYSLOG\n");

```

```

elseif infi.action_type==64 then
    lp.Print(" ACTIONTYPE_TRAP\n");
elseif infi.action_type==65 then
    lp.Print(" ACTIONTYPE_CUSTOM");
elseif infi.action_type==66 then
    lp.Print(" ACTIONTYPE_SEARCH");
elseif infi.action_type==67 then
    lp.Print(" ACTIONTYPE_NULL");
elseif infi.action_type==68 then
    lp.Print(" ACTIONTYPE_ROUTE_EVENT");
elseif infi.action_type==69 then
    lp.Print(" ACTIONTYPE_EVENT");
end

    lp.Print(string.format("\tstring [%s]\n",infi.string));

--[
#define TRAP_FILTER_VALUE_NONE 0
#define TRAP_FILTER_VALUE_CONTAIN_CASE 1
#define TRAP_FILTER_VALUE_CONTAIN_NOCASE 2
#define TRAP_FILTER_VALUE_EGAL_CASE 3
#define TRAP_FILTER_VALUE_EGAL_NOCASE 4
#define TRAP_FILTER_VALUE_SUP 5
#define TRAP_FILTER_VALUE_INF 6
#define TRAP_FILTER_VALUE_DIF 7
#define TRAP_FILTER_VALUE_SUP_EGAL 8
#define TRAP_FILTER_VALUE_INF_EGAL 9
--]]

    lp.Print(string.format("\tcounter %d\n",infi.counter));

    lp.Print(string.format("\tlast %d\n",infi.last));
    lp.Print(string.format("\tfirst %d\n",infi.first));
    lp.Print(string.format("\tfirst_since %d\n",infi.first_since));
    lp.Print(string.format("\t number_total %d\n",infi.number_total));
    lp.Print(string.format("\t number_ack %d\n",infi.number_ack));
    lp.Print(string.format("\t number_delete %d\n",infi.number_delete));
    lp.Print(string.format("\t number_auto_delete %d\n",infi.number_auto_delete));
    lp.Print(string.format("\t number_since_clear %d\n",infi.number_since_clear));
    lp.Print(string.format("\t number_not_displayed %d\n",infi.number_not_displayed));
    lp.Print(string.format("\t number_displayed %d\n",infi.number_displayed));
    lp.Print(string.format("\t max_display %d\n",infi.max_display));
    lp.Print(string.format("\t number_between_action %d\n",infi.number_between_action));
    lp.Print(string.format("\t max_action %d\n",infi.max_action));
    lp.Print(string.format("\t max_display %d\n",infi.max_display));
    lp.Print(string.format("\t number_between_action %d\n",infi.number_between_action));
    lp.Print(string.format("\t action_threshold %d\n",infi.action_threshold));
    lp.Print(string.format("\t action_mode %d\n",infi.action_mode));
    lp.Print(string.format("\t in_time %d\n",infi.in_time));
    lp.Print(string.format("\t clock %d\n",infi.clock));

end
end
end

    end
end
end

```


Ip.GetTrapFilterInformation

```
value=Ip.GetTrapFilterInformation('uid','Array')
```

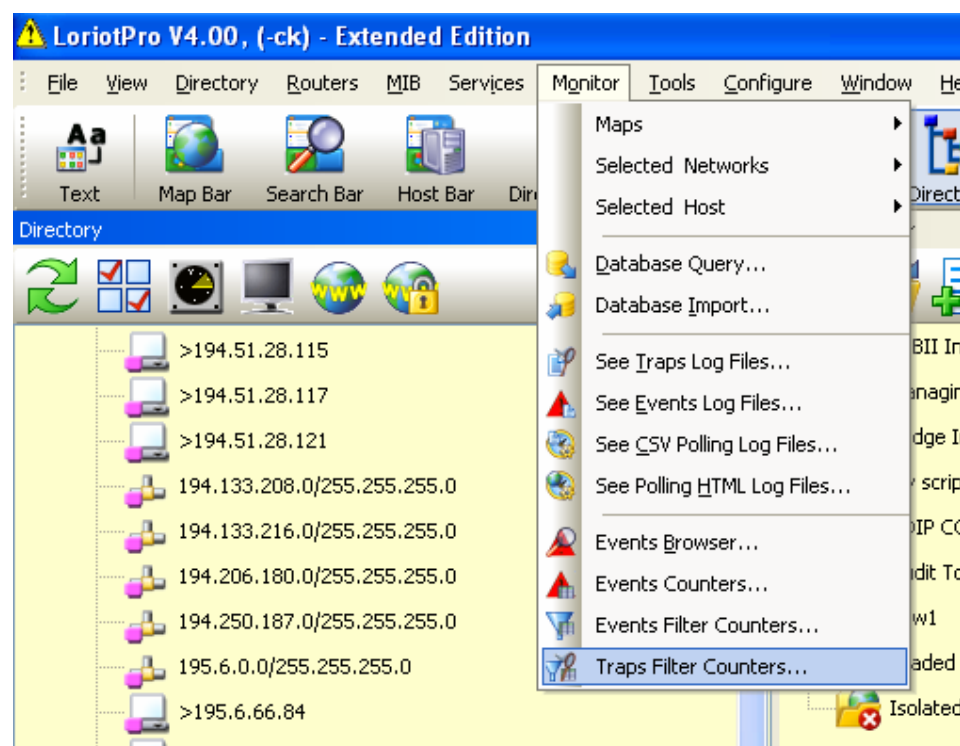
Explication

Cette fonction retourne une array contenant les paramètres associés au trap référencé par l'UID.

Paramètres en entrée

'uid' : l'uid du trap dont on veut collecter les paramètres.

Vous pouvez voir la liste des UID associés au trap avec le module trap filter counters. Attention la colonne UID est cachée (faite glisser avec la souris la limite de colonne de 'Total filtered' pour faire apparaître la colonne UID).



Reference	UID	Total Filtered	Matched Filters Since Last Counter Cl..	Last
>LinkDown[2-0]	4839452334612283392	0	0	
>Authentication[4-0]	4839452334612283394	3493	3493	Tue Jul 04 15:11:01 2006
>LinkUp[3-0]	4839452334612283395	0	0	
>ccmMIBNotificationPrefix [6-6]	4899299538491146252	0	0	
>microsoft.1.1.3.1.2 [6-0]	4935959274795827217	0	0	
>cpqDsCpuTimeFailed_v1 [6-19002]	4935952547907469331	0	0	
>cpqMeFallingAlarmExtended_v1 [6-1]	4936005888075890710	0	0	

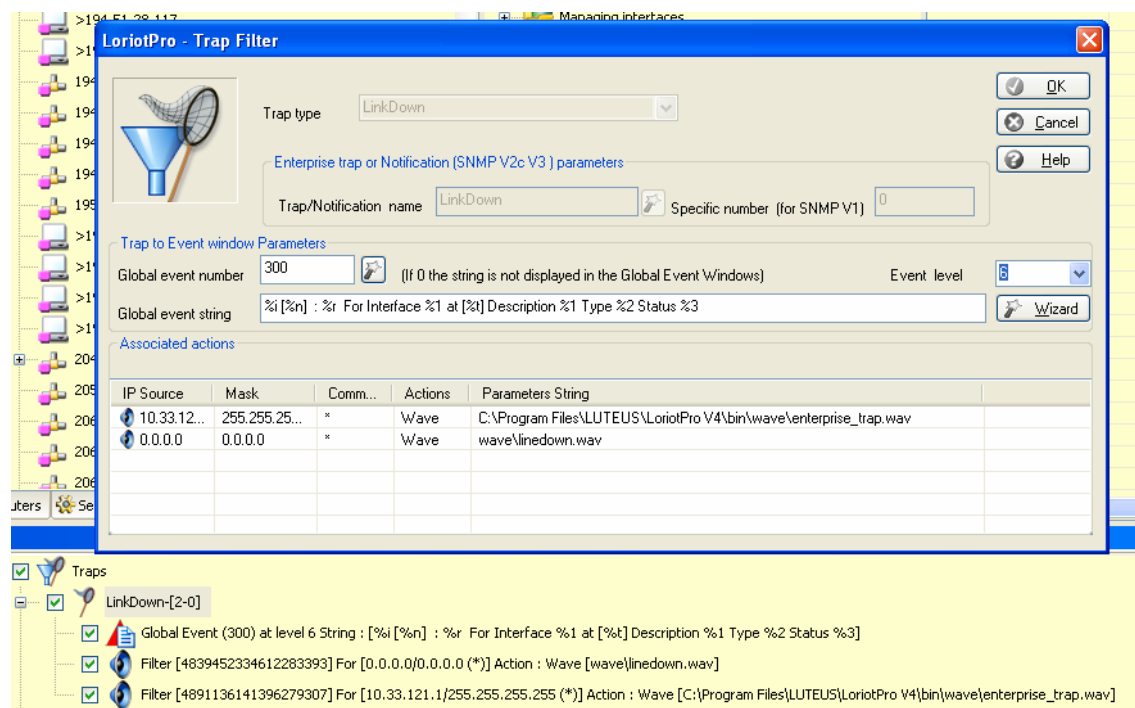
Reference	UID	Total Filtered	Matched Filters Since Last Counter Cl..	Last
>LinkDown[2-0]	4839452334612283392	0	0	
LinkDown	4839452334612283393	0	0	
LinkDown	4891136141396279307	0	0	
>Authentication[4-0]	4839452334612283394	3524	3524	Tue Jul 04 15:13:35 2006
>LinkUp[3-0]	4839452334612283395	0	0	
>ccmMIBNotificationPrefix [6-6]	4899299538491146252	0	0	
>microsoft.1.1.3.1.2 [6-0]	4935959274795827217	0	0	
>cpqDsCpuTimeFailed_v1 [6-19002]	4935952547907469331	0	0	
compaq	4935952870030016532	0	0	
>cpqMeFallingAlarmExtend	4936005888075890710	0	0	
compaq	4936007258170458135	0	0	

'array' : l'array pour collecter les paramètres associés au trap filter.

Value = nil si pas d'information

Valeur	Description
Array.trap	Le nom du trap associé a l'uid
Array.name	
Array.specific	Le numero specific du trap si trap V1
Array.eventnumber	Le numero d'Event associé au trap et rerouté vers le fenêtre 'Global Event'
Array.string	La chaine de caractères envoyée au 'global event' construite à partir des paramètres du trap
Array.ctrl	Une valeur de contrôle 0xFADA
Array.counter	Le nombre de trap reçu depuis le départ de LoriotPro
Array.uid	L'uid du trap
Array.last	Un timestamp du dernier trap de ce type reçu.
Array.first	Un timestamp du premier trap de ce type reçu.
Array.first_since	Un timestamp du premier trap de ce type reçu depuis un aquitement global.

Array.number_total	Le nombre total de trap reçu
Array.number_ack	Le nombre de trap acquitté
Array.number_delete	Le nombre de trap effacé
Array.number_auto_delete	Le nombre de trap effacé automatiquement car le listbox de reception est plein.
Array.number_since_clear	Le nombre reçût depuis un acquittement global.
Array.number_not_displayed	Nombre nom affiché dans le listbox du 'workspace'
Array.number_displayed	Nombre affiché dans le listbox du 'workspace'
Array.max_display	Le nombre maximum de trap à affiché dans le listbox (-1 egal pas de limit)
Array.number_between_action	Usage réservé
Array.disable	Etat du filtre 1 - Ce filtre est non actif (les actions associées également) 0 – ce filtre est actif



Exemple :

```

if Ip.GetTrapFilterInformation(ta[j],"inf") then
Ip.Print(string.format("Trap \t%s\n",inf.trap));
Ip.Print(string.format("\tTrap V1 Name \t%s\n",inf.name));
Ip.Print(string.format("\tTrap V1 Generic %i Specific %i\n",inf.generic,inf.specific));
Ip.Print(string.format("\tAssociated event number %d\n",inf.eventnumber));
    
```

```
lp.Print(string.format("\tAssociated string [%s]\n",inf.string));
lp.Print(string.format("\tStructure controle 0x%X\n",inf.ctrl));
lp.Print(string.format("\tcounter %d\n",inf.counter));
lp.Print(string.format("\tuid %s\n",inf.uid));

lp.Print(string.format("\tlast %d\n",inf.last));
lp.Print(string.format("\tfirst %d\n",inf.first));
lp.Print(string.format("\tfirst_since %d\n",inf.first_since));

lp.Print(string.format("\t number_total %d\n",inf.number_total));
lp.Print(string.format("\t number_ack %d\n",inf.number_ack));
lp.Print(string.format("\t number_delete %d\n",inf.number_delete));
lp.Print(string.format("\t number_auto_delete %d\n",inf.number_auto_delete));
lp.Print(string.format("\t number_since_clear %d\n",inf.number_since_clear));
lp.Print(string.format("\t number_not_displayed %d\n",inf.number_not_displayed));
lp.Print(string.format("\t number_displayed %d\n",inf.number_displayed));
lp.Print(string.format("\t max_display %d\n",inf.max_display));
lp.Print(string.format("\t number_between_action %d\n",inf.number_between_action));
lp.Print(string.format("\t max_action %d\n",inf.max_action));
lp.Print(string.format("\t max_display %d\n",inf.max_display));
lp.Print(string.format("\t number_between_action %d\n",inf.number_between_action));
end
```

Ip.GetTrapFilterActionList

```
number=Ip.GetTrapFilterActionList('uid', 'Array');
```

Explication

Cette fonction permet de collecter la liste des filtres (uid) associés à un trap (uid).

Paramètres en entrée

'uid' : l'uid d'un trap filter

voir la fonction "**Ip.GetTrapFilterInformation** » pour obtenir l'uid associé a un filtre de trap.

'array' :

un tableau de collecte des resultats, retourne l'uid des actions

array[0]... array[number-1]

Exemple

```
number2=Ip.GetTrapFilterActionList(ta[j],"taa");

if number2 then
Ip.Print(string.format("\nFound %i Action registered for this trap filter\n\n",number2));
for jj=0,(number2-1) do

Ip.Print(string.format("\tAck Action UID [%s] for  %s %s\n",taa[jj],ta[j],inf.trap));
Ip.AckTrapFilterAction(taa[jj]);
    if Ip.GetTrapFilterActionInformation(taa[jj],"infi") then
        Ip.Print(string.format("\tif this parameter match : \tip \t%s mask \t %s ",infi.ip,infi.mask));
        Ip.Print(string.format("\tcommunity \t%s\n",infi.community));
        Ip.Print(string.format("\tSearch string value 0 [%s] condition [%i]
\n",infi.search_string0,infi.search_condition0));
        Ip.Print(string.format("\tSearch string value 1 [%s] condition [%i]
\n",infi.search_string1,infi.search_condition1));
        Ip.Print(string.format("\tSearch string value 2 [%s] condition [%i]
\n",infi.search_string2,infi.search_condition2));
        Ip.Print(string.format("\tSearch string value 3 [%s] condition [%i]
\n",infi.search_string3,infi.search_condition3));
        Ip.Print(string.format("\tSearch string value 4 [%s] condition [%i]
\n",infi.search_string4,infi.search_condition4));
        Ip.Print(string.format("\tSearch string value 5 [%s] condition [%i]
\n",infi.search_string5,infi.search_condition5));
        Ip.Print(string.format("\tSearch string value 6 [%s] condition [%i]
\n",infi.search_string6,infi.search_condition6));
        Ip.Print(string.format("\tSearch string value 7 [%s] condition [%i]
\n",infi.search_string7,infi.search_condition7));
        Ip.Print(string.format("\tSearch string value 8 [%s] condition [%i]
\n",infi.search_string8,infi.search_condition8));
        Ip.Print(string.format("\tSearch string value 9 [%s] condition [%i]
\n",infi.search_string9,infi.search_condition9));
```



```

    lp.Print(string.format("\taction_type %d ",infi.action_type));

--[
#define ACTIONTYPE_WINRUN 1
#define ACTIONTYPE_WAVE 2
#define ACTIONTYPE_SMTMP 4
#define ACTIONTYPE_MAPI 8
#define ACTIONTYPE_DOSRUN 16
#define ACTIONTYPE_SYSLOG 32
#define ACTIONTYPE_TRAP 64
#define ACTIONTYPE_CUSTOM 65
#define ACTIONTYPE_SEARCH 66
#define ACTIONTYPE_NULL 67
#define ACTIONTYPE_ROUTE_EVENT 68
//v4
#define ACTIONTYPE_EVENT 69
--]]
    if infi.action_type==1 then
        lp.Print(" ACTIONTYPE_WINRUN");
    elseif infi.action_type==2 then
        lp.Print(" ACTIONTYPE_WAVE");
    elseif infi.action_type==4 then
        lp.Print(" ACTIONTYPE_SMTMP");
    elseif infi.action_type==8 then
        lp.Print(" ACTIONTYPE_MAPI");
    elseif infi.action_type==16 then
        lp.Print(" ACTIONTYPE_DOSRUN");
    elseif infi.action_type==32 then
        lp.Print(" ACTIONTYPE_SYSLOG\n");
    elseif infi.action_type==64 then
        lp.Print(" ACTIONTYPE_TRAP\n");
    elseif infi.action_type==65 then
        lp.Print(" ACTIONTYPE_CUSTOM");
    elseif infi.action_type==66 then
        lp.Print(" ACTIONTYPE_SEARCH");
    elseif infi.action_type==67 then
        lp.Print(" ACTIONTYPE_NULL");
    elseif infi.action_type==68 then
        lp.Print(" ACTIONTYPE_ROUTE_EVENT");
    elseif infi.action_type==69 then
        lp.Print(" ACTIONTYPE_EVENT");
    end

    lp.Print(string.format("\tstring [%s]\n",infi.string));

--[
#define TRAP_FILTER_VALUE_NONE 0
#define TRAP_FILTER_VALUE_CONTAIN_CASE 1
#define TRAP_FILTER_VALUE_CONTAIN_NOCASE 2
#define TRAP_FILTER_VALUE_EQUAL_CASE 3
#define TRAP_FILTER_VALUE_EQUAL_NOCASE 4
#define TRAP_FILTER_VALUE_SUP 5
#define TRAP_FILTER_VALUE_INF 6
#define TRAP_FILTER_VALUE_DIF 7
#define TRAP_FILTER_VALUE_SUP_EQUAL 8
#define TRAP_FILTER_VALUE_INF_EQUAL 9
--]]

    lp.Print(string.format("\tcounter %d\n",infi.counter));

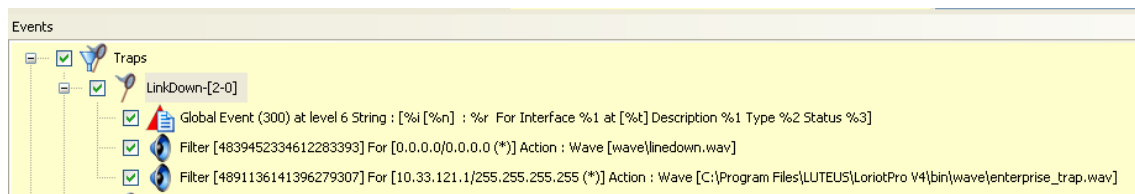
```

```

lp.Print(string.format("\tlast %d\n",infi.last));
lp.Print(string.format("\tfirst %d\n",infi.first));
lp.Print(string.format("\tfirst_since %d\n",infi.first_since));
lp.Print(string.format("\t number_total %d\n",infi.number_total));
lp.Print(string.format("\t number_ack %d\n",infi.number_ack));
lp.Print(string.format("\t number_delete %d\n",infi.number_delete));
lp.Print(string.format("\t number_auto_delete %d\n",infi.number_auto_delete));
lp.Print(string.format("\t number_since_clear %d\n",infi.number_since_clear));
lp.Print(string.format("\t number_not_displayed %d\n",infi.number_not_displayed));
lp.Print(string.format("\t number_displayed %d\n",infi.number_displayed));
lp.Print(string.format("\t max_display %d\n",infi.max_display));
lp.Print(string.format("\t number_between_action %d\n",infi.number_between_action));
lp.Print(string.format("\t max_action %d\n",infi.max_action));
lp.Print(string.format("\t max_display %d\n",infi.max_display));
lp.Print(string.format("\t number_between_action %d\n",infi.number_between_action));
lp.Print(string.format("\t action_threshold %d\n",infi.action_threshold));
lp.Print(string.format("\t action_mode %d\n",infi.action_mode));
lp.Print(string.format("\t in_time %d\n",infi.in_time));
lp.Print(string.format("\t clock %d\n",infi.clock));

```

end



Ip.GetTrapFilterActionInformation

```
value=Ip.GetTrapFilterActionInformation('uid','Array')
```

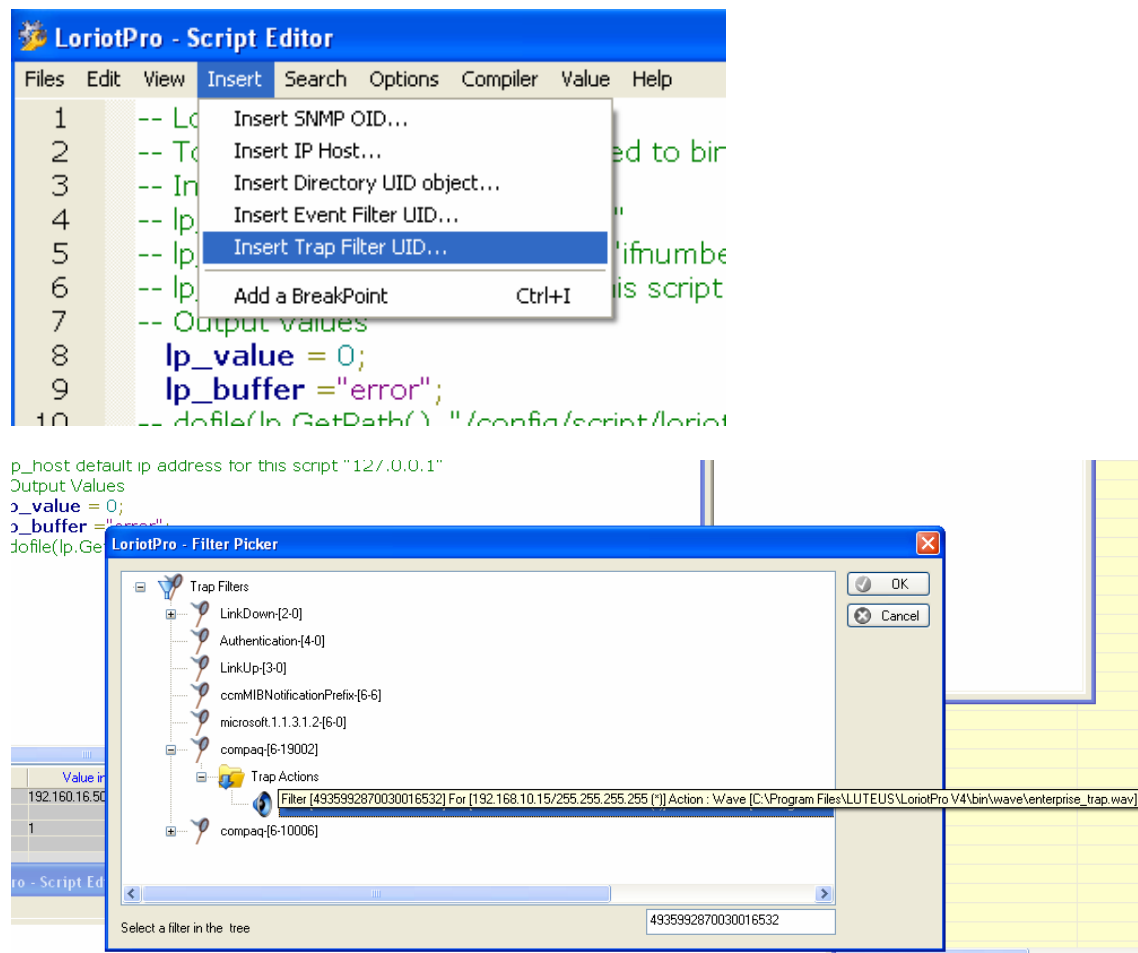
Explication

Cette fonction retourne un tableau contenant les paramètres du filtre (uid) associé à un trap.

Paramètres en entrée

'uid' : l'uid de l'action associé au trap filter

Un wizard est fourni avec l'éditeur de script pour vous aider à trouver un uid d'action de trap filter.

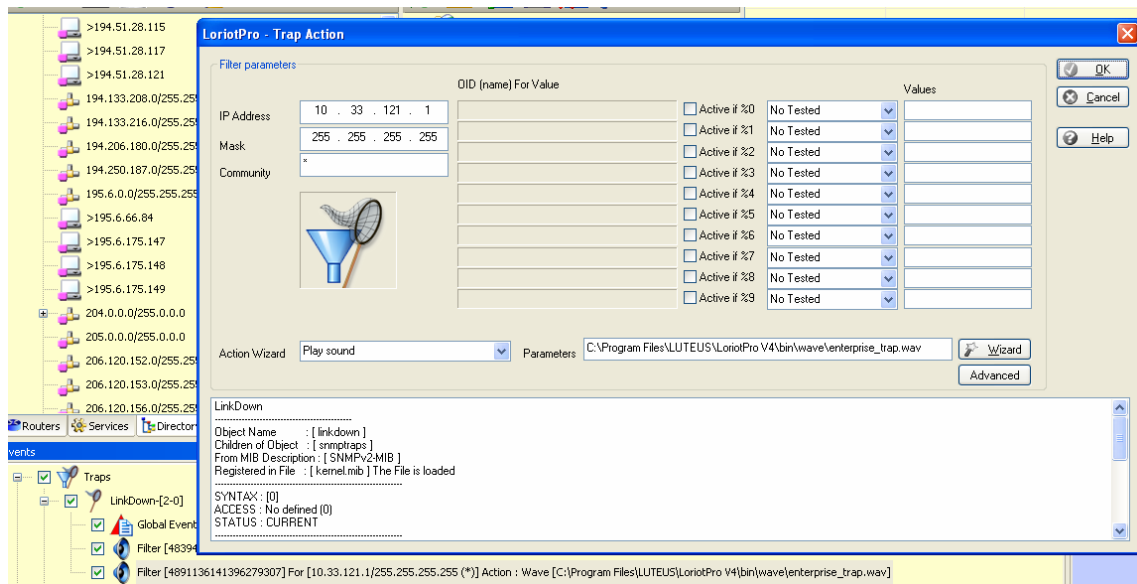


'array' : un tableau contenant les réponses

Valeur	Description
Array.ip	L'adresse ip du filtre

Array.mask	Le mask pour l'adresse IP
Array.community	La community contenue dans le paquet reçu (* = all)
Array.search_string0	Chaine de caractère utilisée avec la variable search_condition pour réaliser un test
Array.search_condition0	0 nom actif 1 contient cette string avec respect de la case 2 contient cette string sans respect de la case 3 contient cette valeur (respect de la case) 4 contient cette valeur (sans test de la case) 5 valeur supérieur a 6 valeur inférieur a 7 valeur différente a 8 valeur supérieur ou egal a 9 valeur inférieur ou egal a
Array.search_string1	idem
Array.search_condition1	idem
Array.search_string2	idem
Array.search_condition2	idem
Array.search_string3	idem
Array.search_condition3	idem
Array.search_string4	idem
Array.search_condition4	idem
Array.search_string5	idem
Array.search_condition5	idem
Array.search_string6	idem
Array.search_condition6	idem
Array.search_string7	idem
Array.search_condition7	idem
Array.search_string8	idem
Array.search_condition8	idem
Array.search_string9	idem
Array.search_condition9	idem
Array.action_type	--[[#define ACTIONTYPE_WINRUN 1 #define ACTIONTYPE_WAVE 2 #define ACTIONTYPE_SMTP 4 #define ACTIONTYPE_MAPI 8 #define ACTIONTYPE_DOSRUN 16 #define ACTIONTYPE_SYSLOG 32 #define ACTIONTYPE_TRAP 64 #define ACTIONTYPE_CUSTOM 65 #define ACTIONTYPE_SEARCH 66 #define ACTIONTYPE_NULL 67 #define ACTIONTYPE_ROUTE_EVENT 68 //v4 #define ACTIONTYPE_EVENT 69 --]]
Array.string	Une cahaine de caractère contenant les paramètres de l'action.

<p>Array.counter Array.last Array.first Array.first_since Array.number_total Array.number_ack Array.number_delete Array.number_auto_delete Array.number_since_clear Array.number_not_displayed Array.number_displayed Array.max_display Array.number_between_action Array.action_threshold Array.action_mode Array.in_time Array.clock Array.disable</p>	<p>Voir : Ip.GetTrapFilterInformation</p>
---	---



Exemple : voir GetTrapFilterActionList

```
if Ip.GetTrapFilterActionInformation(taa[jjj],"infi") then
```

Ip.GetEvenList

```
number=Ip.GetEventList('array');
```

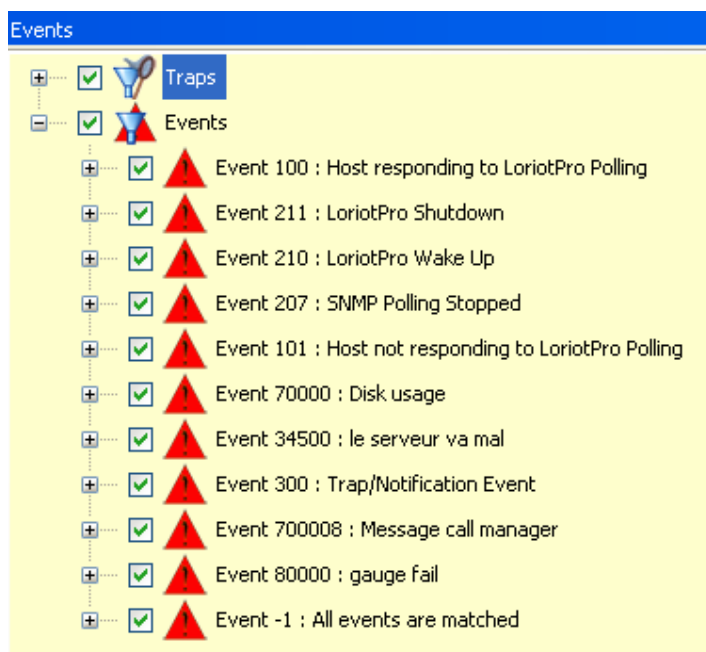
Explication

Cette fonction retourne la liste dans un tableau des événements (number) référencé.

Paramètres en entrée

'array' : un tableau contenant les résultats de la requête.

array[0]...array[number-1]
l'array[0] contient un numero d'événement



```
number=Ip.GetEventList("ta");

if number then
Ip.Print(string.format("\nFound %i Loriot Event registered\n",number));
for j=0,(number-1) do
Ip.Print(string.format("Event Reference %s\n",ta[j]));
Ip.Print("Ack -----\n");
Ip.AckEvent(ta[j]);

    if Ip.GetEventInformation(ta[j],"inf") then
Ip.Print(string.format("Loriot Event : %d \t%s\n",ta[j],inf.description));
Ip.Print(string.format("\tlast %d\n",inf.last));
Ip.Print(string.format("\tfirst %d\n",inf.first));
Ip.Print(string.format("\tfirst_since %d\n",inf.first_since));

Ip.Print(string.format("\t number_total %d\n",inf.number_total));
```

```

lp.Print(string.format("\t number_ack %d\n",inf.number_ack));
lp.Print(string.format("\t number_delete %d\n",inf.number_delete));
lp.Print(string.format("\t number_auto_delete %d\n",inf.number_auto_delete));
lp.Print(string.format("\t number_since_clear %d\n",inf.number_since_clear));
lp.Print(string.format("\t number_not_displayed %d\n",inf.number_not_displayed));
lp.Print(string.format("\t number_displayed %d\n",inf.number_displayed));
lp.Print(string.format("\t max_display %d\n",inf.max_display));
lp.Print(string.format("\t number_between_action %d\n",inf.number_between_action));
lp.Print(string.format("\t max_action %d\n",inf.max_action));
lp.Print(string.format("\t max_display %d\n",inf.max_display));
lp.Print(string.format("\t number_between_action %d\n",inf.number_between_action));

number2=lp.GetEventActionList(ta[jj],"taa");

if number2 then
lp.Print(string.format("\nFound %i filter/Action registered for this event\n\n",number2));

for jj=0,(number2-1) do
lp.Print(string.format("\tAck Action UID [%s] for %s\n",taa[jj],ta[jj]));
lp.AckEventFilterAction(taa[jj]);
  if lp.GetEventFilterActionInformation(taa[jj],"infi") then
    lp.Print(string.format("\tref %s ",infi.ref));
    lp.Print(string.format("\tip %s mask %s\n",infi.ip,infi.mask));
    lp.Print(string.format("\tSearch string value 0 [%s] condition [%i]
\n",infi.search_string0,infi.search_condition0));
    lp.Print(string.format("\tSearch string value 1 [%s] condition [%i]
\n",infi.search_string1,infi.search_condition1));

    lp.Print(string.format("\taction_type %d ",infi.action_type));
    if inf.action_type==1 then
      lp.Print(" ACTIONTYPE_WINRUN");
    elseif inf.action_type==2 then
      lp.Print(" ACTIONTYPE_WAVE");
    elseif inf.action_type==4 then
      lp.Print(" ACTIONTYPE_SMTP");
    elseif inf.action_type==8 then
      lp.Print(" ACTIONTYPE_MAPI");
    elseif inf.action_type==16 then
      lp.Print(" ACTIONTYPE_DOSRUN");
    elseif inf.action_type==32 then
      lp.Print(" ACTIONTYPE_SYSLOG");
    elseif inf.action_type==64 then
      lp.Print(" ACTIONTYPE_TRAP");
    elseif inf.action_type==65 then
      lp.Print(" ACTIONTYPE_CUSTOM");
    elseif inf.action_type==66 then
      lp.Print(" ACTIONTYPE_SEARCH");
    elseif inf.action_type==67 then
      lp.Print(" ACTIONTYPE_NULL");
    elseif inf.action_type==68 then
      lp.Print(" ACTIONTYPE_ROUTE_EVENT");
    elseif inf.action_type==69 then
      lp.Print(" ACTIONTYPE_EVENT");
    end

    lp.Print(string.format("\tstring [%s]\n",infi.string));

lp.Print(string.format("\tcounter %d\n",infi.counter));

```

```
lp.Print(string.format("\tlast %d\n",infi.last));
lp.Print(string.format("\tfirst %d\n",infi.first));
lp.Print(string.format("\tfirst_since %d\n",infi.first_since));
lp.Print(string.format("\t number_total %d\n",infi.number_total));
lp.Print(string.format("\t number_ack %d\n",infi.number_ack));
lp.Print(string.format("\t number_delete %d\n",infi.number_delete));
lp.Print(string.format("\t number_auto_delete %d\n",infi.number_auto_delete));
lp.Print(string.format("\t number_since_clear %d\n",infi.number_since_clear));
lp.Print(string.format("\t number_not_displayed %d\n",infi.number_not_displayed));
lp.Print(string.format("\t number_displayed %d\n",infi.number_displayed));
lp.Print(string.format("\t max_display %d\n",infi.max_display));
lp.Print(string.format("\t number_between_action %d\n",infi.number_between_action));
lp.Print(string.format("\t max_action %d\n",infi.max_action));
lp.Print(string.format("\t max_display %d\n",infi.max_display));
lp.Print(string.format("\t number_between_action %d\n",infi.number_between_action));
lp.Print(string.format("\t action_threshold %d\n",infi.action_threshold));
lp.Print(string.format("\t action_mode %d\n",infi.action_mode));
lp.Print(string.format("\t in_time %d\n",infi.in_time));
lp.Print(string.format("\t clock %d\n",infi.clock));

end

end
end
  end
end
end
```


Ip.GetEventInformation

```
Ip.GetEventInformation(event_number,'array');
```

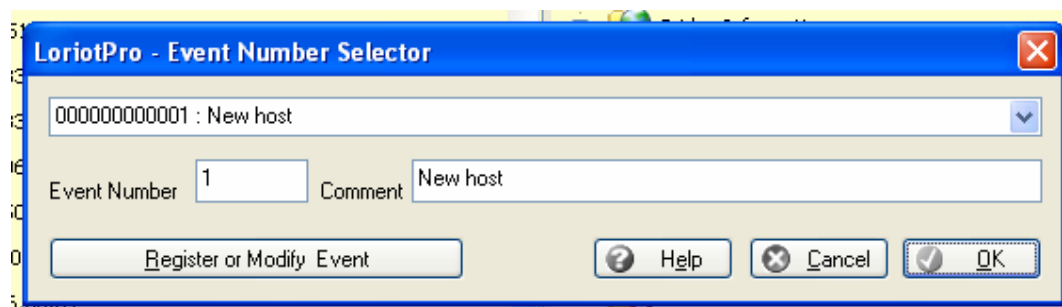
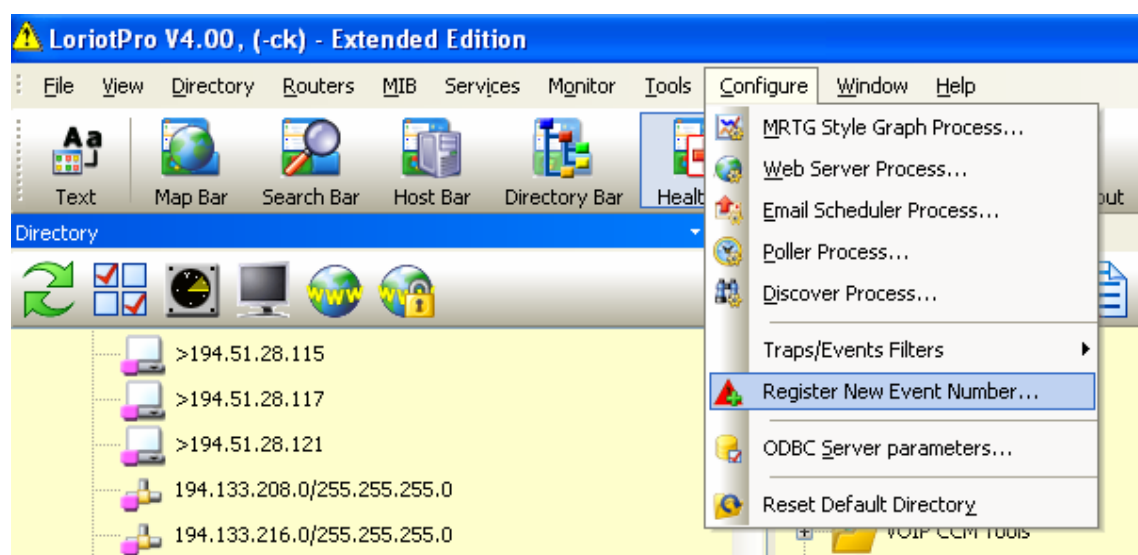
Explication

Cette fonction permet de collecter dans un tableau les paramètres d'un numéro d'événement reference.

Paramètres en entrée

Event_number : un numero d'événement

Le fichier bin/config/events.txt contient la liste des numéros d'événement enregistrés par le système.



'array' : un tableau contenant les paramètres associées

Valeur	Description
Array.description	La description associée à l'événement

Array.last	Un timestamp du dernier événement de ce type reçu.
Array.first	Un timestamp du premier événement de ce type reçu.
Array.first_since	Un timestamp du premier événement de ce type reçu depuis un acquittement global.
Array.number_total	Le nombre total des événements reçus
Array.number_ack	Le nombre d'événements acquittés
Array.number_delete	Le nombre d'événements effacés
Array.number_auto_delete	Le nombre d'événements effacés automatiquement car le listbox de réception est pleine.
Array.number_since_clear	Le nombre reçu depuis un acquittement global.
Array.number_not_displayed	Nombre d'événements non affichés dans le listbox du 'workspace'
Array.number_displayed	Nombre total des événements qui ont été affichés dans le listbox du 'workspace'
Array.max_display	Le nombre maximum d'événements à afficher dans le listbox (-1 égal pas de limit)
Array.number_between_action	Usage réservé
Array.disable	Etat du filtre 1 - Ce filtre est non actif (les actions associées également) 0 – ce filtre est actif

Exemple : voir la fonction `Ip.GetEventList('array')`;

Ip.GetEventActionList

```
number= Ip.GetEventActionList(event_number,'array');
```

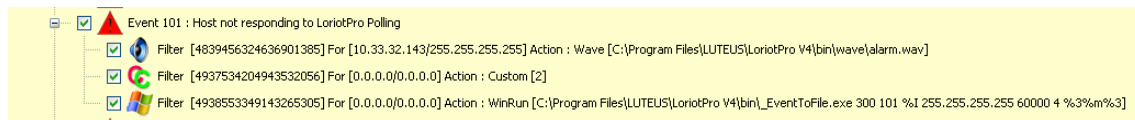
Explication

Cette fonction retourne la liste (uid) des filtres d'actions associée à un numéro d'événement.

Paramètres en entrée

Event_number : le numéro d'événement associé a la liste d'action
'array' : un tableau contenant la liste des actions associées a l'événement.

array[0]..array[number-1]
array[x] contient l'uid de l'action



Exemple : voir la fonction Ip.GetEventList('array');

Ip.GetEventFilterActionInformation

```
value=Ip.GetEventFilterActionInformation('uid','Array')
```

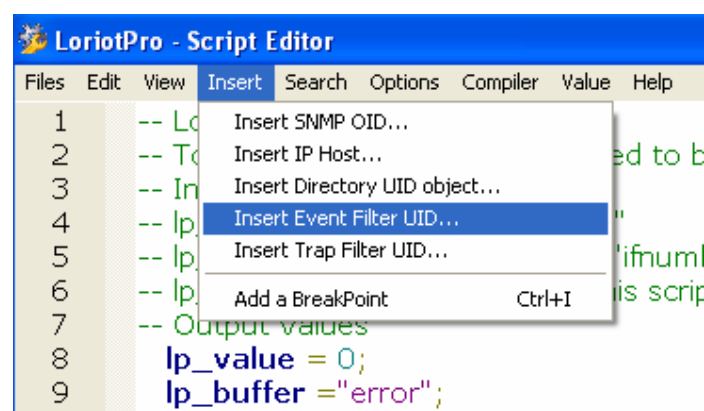
Explication

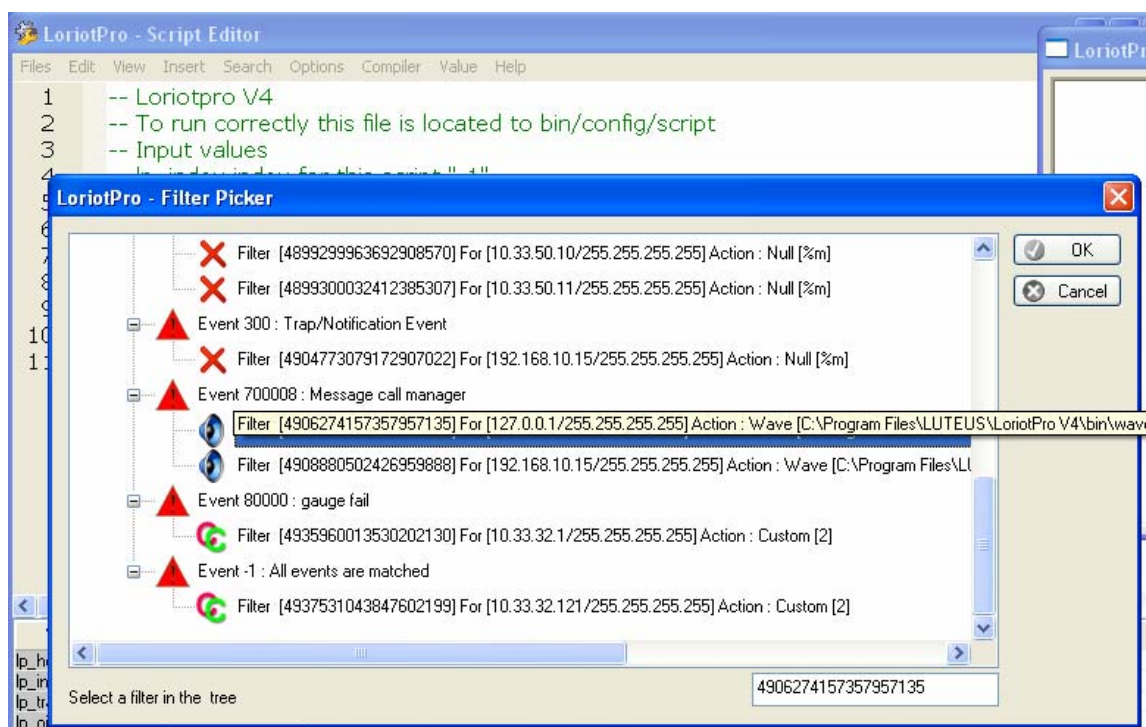
Cette fonction retourne un tableau avec les parametres associés à un filtre d'événement.

Paramètres en entrée

'uid' : l'uid de l'action associée au filtre d'événement

Un wizard est fourni dans l'éditeur de script pour vous aider a trouver l'uid d'une action de filtre d'événements :

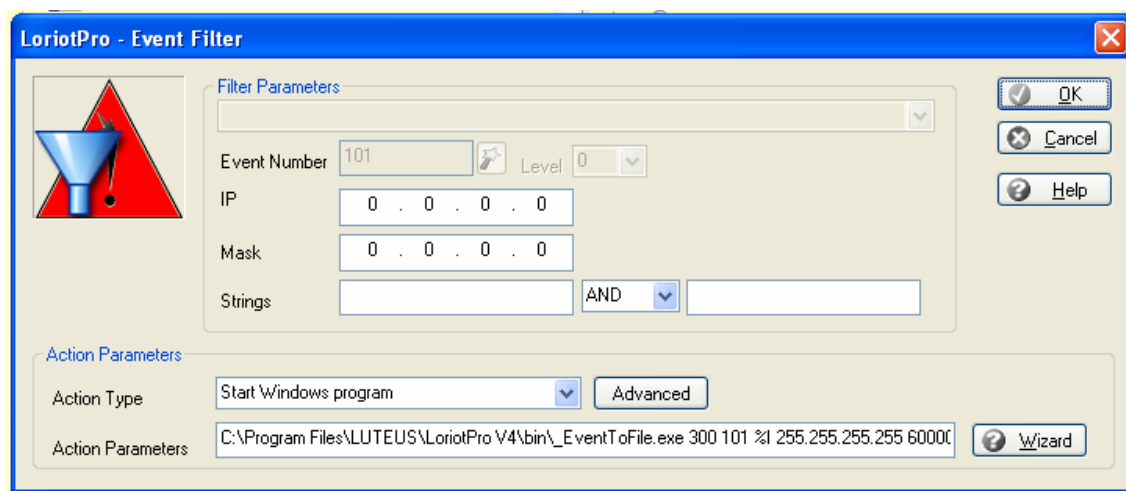




'array' : un tableau contenant les paramètres de l'action

Valeur	Description
Array.ref	Le numéro d'événement associé a l'action
Array.ip	L'adresse ip du filtre
Array.mask	Le mask pour l'adresse IP
Array.search_string0	Chaine de caractères utilisée avec la variable search_condition pour réaliser un test
Array.search_condition0	Reservé
Array.search_string1	idem
Array.search_condition1	idem
Array.action_type	--[[#define ACTIONTYPE_WINRUN 1 #define ACTIONTYPE_WAVE 2 #define ACTIONTYPE_SMTP 4 #define ACTIONTYPE_MAPI 8 #define ACTIONTYPE_DOSRUN 16 #define ACTIONTYPE_SYSLOG 32 #define ACTIONTYPE_TRAP 64 #define ACTIONTYPE_CUSTOM 65 #define ACTIONTYPE_SEARCH 66

	<pre>#define ACTIONTYPE_NULL 67 #define ACTIONTYPE_ROUTE_EVENT 68 //v4 #define ACTIONTYPE_EVENT 69 --]]</pre>
Array.string	Une chaîne de caractères contenant les paramètres de l'action.
<p>Array.counter</p> <p>Array.last</p> <p>Array.first</p> <p>Array.first_since</p> <p>Array.number_total</p> <p>Array.number_ack</p> <p>Array.number_delete</p> <p>Array.number_auto_delete</p> <p>Array.number_since_clear</p> <p>Array.number_not_displayed</p> <p>Array.number_displayed</p> <p>Array.max_display</p> <p>Array.number_between_action</p> <p>Array.action_threshold</p> <p>Array.action_mode</p> <p>Array.in_time</p> <p>Array.clock</p> <p>Array.disable</p>	Voir : Ip.GetEventFilterInformation



Exemple : voir la fonction Ip.GetEventList('array');

Ip.AckEvent

Ip.AckEvent(event_number)

Explication

Cette fonction permet de faire un acquittement « ack » des événements

Paramètres en entrée

event_number : le numero d'événement à acquitter

ATTENTION :

Tous les événements de référence 'event_number' seront acquittés sans tenir compte des paramètres associés.

Ip.AckEventFilterAction

```
Ip.AckEventFilterAction('uid');
```

Explication

Cette fonction permet de faire un ack d'un filtre d'événement (uid)

Ip.AckTrapFilter

```
Ip.AckTrapFilter('uid');
```

Explication

Cette fonction permet de faire un ack d'un type de trap reçu (uid).

ATTENTION :

Tous les trap de référence 'uid' seront acquittés sans tenir compte des paramètres associés. DAns le cas d'un linkdown pae exemple tous les linkdown seront acquittés.

Ip.AckTrapFilterAction

```
Ip.AckTrapFilterAction('uid');
```

Explication

Cette fonction permet de faire un ack d'un filtre de trap (uid)

```
Ip.ClearIPEventCounter
```

```
Ip.ClearIPEventCounter('ip');
```

Explication

Cette fonction permet de remettre à zero le compteur d'événements reçus pour un host.

Ip.ClearIPTrapCounter

```
Ip.ClearIPTrapCounter('ip');
```

Explication

Cette fonction permet de remettre a zero le compteur de traps reçus pour un host.

Ip.SetEventFilterActionDisable

```
Ip.SetEventFilterActionDisable('uid',0/1) ;
```

Explication

Cette fonction permet d'activer ou de désactiver dynamiquement un filtre d'événements.

Paramètres en entrée

'uid' : l'uid de l'action associé à un événement

0 active

1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.SetTrapFilterActionDisable

```
Ip.SetTrapFilterActionDisable('uid',0/1) ;
```

Explication

Cette fonction permet d'activer ou de désactiver dynamiquement un filtre de trap.

Paramètres en entrée

'uid' : l'uid de l'action associé à un trap

0 active

1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.SetEventFilterDisable

Ip.SetEventFilterDisable(event_number)

Explication

Cette fonction permet d'activer ou de désactiver dynamiquement tous les filtres associés à un numéro d'événement.

Paramètres en entrée

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.SetTrapFilterDisable

Ip.SetTrapFilterDisable('uid',0/1)

Explication

Cette fonction permet d'activer ou de désactiver dynamiquement tous les filtres associés à un trap (uid).

Paramètres en entrée

0 active
1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.GetTrapFilterGlobalStatus

Ip.GetTrapFilterGlobalStatus()

Explication

Cette fonction permet de récupérer le status global d'activité des filtres de trap

Paramètres en entrée

0 active
1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.GetEventFilterGlobalStatus

Ip.GetEventFilterGlobalStatus()

Explication

Cette fonction permet de récupérer le status global d'activité des filtres d'événement

Paramètres en entrée

0 active
1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.SetTrapFilterGlobalStatus

Ip.SetTrapFilterGlobalStatus(0/1)

Explication

Cette fonction permet de définir le status global d'activité des filtres de trap

Paramètres en entrée

0 active
1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.SetEventFilterGlobalStatus

Ip.SetEventFilterGlobalStatus(0/1)

Explication

Cette fonction permet de définir le status global d'activité des filtres d'événement

Paramètres en entrée

0 active
1 désactive

ATTENTION :

la case à cocher de l'arbre des filtres n'est pas automatiquement rafraichie.

Ip.GetLoriotProID

string=Ip.GetLoriotProID()

Explication

Cette fonction retourne l'ID de LoriotPro

LoriotPro ActiveView (Ipav) Librairie

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 (vectoriel) 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'objet qui deviennent « 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 que de travailler 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'un ActiveView est séquentiel, les scripts se dérouleront donc les un après les autres. L'ordre de réalisation des scripts dépend de la liste chaînée des objets en mémoire.

lpav.Insert

```
result=lpav.Insert(x,y,cx,cy,type);
```

Explication

result=lpav.Insert(x,y,cx,cy,type); permet d'insérer dans l'ActiveView un nouvel objet qui deviendra l'objet par défaut.

Paramètres en entrée

X la position x de l'objet graphique dans l'ActiveView, la valeur doit être positive.
Y la position y de l'objet graphique dans l'ActiveView, la valeur doit être positive.
Cx la largeur de l'objet graphique, la valeur doit être positive.
Cy la hauteur de l'objet graphique, la valeur doit être positive.

type le type d'objet à insérer, la liste est fourni dans le fichier loriotinit.lua en fonction du type choisi LoriotPro intégrera automatiquement l'iconographie associé.

```
type
--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;
LPAV_OBJ_HOST_HUB = 46;
LPAV_OBJ_CONNECTOR = 144;
```

Paramètres en sortie

Result =1 si l'objet est inséré sinon nil

Exemple

L'exemple est complexe et utilise d'autres fonctions mais il est explicite.

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

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,20 do
    lpav.Insert(10*i,1,9,9,zz);
    if (k==0) then lpav.SetRef(1,"r"); k=1; else lpav.SetRef(1,"s"); k=0; end
    lpav.SetBrush(0,255,0);
    lpav.SetFontName("Mistral");
    lpav.SetName(i);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
lpav.SetOption(LPAV_UNIT,"%");
end
zz=LPAV_OBJ_FILL_CIRCLE;
-- Créé des objects s r
for i=0,20 do
    lpav.Insert(10*i,30,9,9,zz);
    if (k==0) then lpav.SetRef(1,"s"); k=1; else lpav.SetRef(1,"r"); k=0; end
    lpav.SetBrush(0,255,0);
    lpav.SetName("test-r"..i);
lpav.SetOption(LPAV_LINESTYLE,55);
end

lpav.ClearAllSelected()
--lpav.DeleteAllSelected();
i=lpav.FindRef(1,0,"r","a");
if (i>0) then
    for j=0, i do
        -- lp.Trace("pass"..j);
        if lpav.SelectUID(a[j]) then
            lpav.SetLineColor(255,45,89);
            lpav.SetLineWidth(3);
        end
    end
end
end
```

```

i=lpav.FindRef(1,0,"s","aa");
if (i>0) then
    for j=0, i do
        --lp.Trace("pass"..j);
        if lpav.SelectUID(aa[j]) then
            lpav.SetLineColor(255,45,255);
            lpav.SetLineWidth(5);
            x,y,x1,y1=lpav.GetPosition();
            lpav.SetPosition(x,y+10,x1,y1);
            lpav.Insert(1,1,100,100,LPAV_OBJ_CONNECTOR);
            lpav.SetOption(LPAV_CONNECTED_TO1,a[j]);
            lpav.SetOption(LPAV_CONNECTED_TO2,aa[j]);
            lpav.SetOption(LPAV_LINEWIDTH,1);
            lpav.SetLineColor(100,45,45);
        end
    end
end
lp.Trace("fin");
lpav.RefreshMap();

```

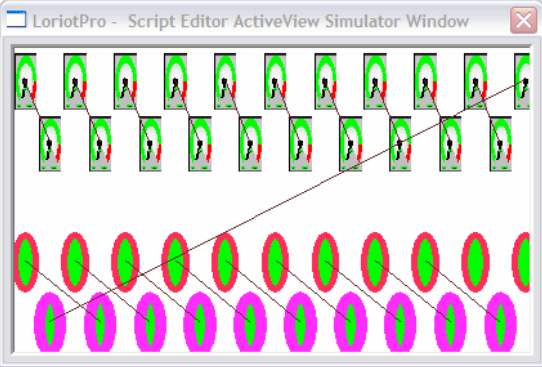
LoriotPro - Script Editor : D:\v4-net\bin\config\script\lpav.insert.lua

Files Edit View Insert Search Options Compiler Value

```

1  -- Loriotpro V4
2  -- Ce fichier doit impérativement se trouver dans bin/config/script
3  -- Parametres passés au script
4  -- lp_index l'index a ajouté au requete ".1"
5  -- lp_oid l'oid snmp passé ou une valeur "ifnumber"
6  -- lp_host l'add ip du host en string "127.0.0.1"
7  -- valeurs à retourner impérativement
8  lp_value = 0;
9  lp_buffer = "error";
10
11 dofile(lp.GetPath().."/config/script/loriotini
12
13 zz=LPAV_OBJ_ANALOGIC_METER_01;
14
15 -- Créé des objets r s
16 k=0;
17 -- for i=0,20 do
18     lpav.Insert(10*i,1,9,9,zz);
19     if (k==0) then lpav.SetRef(1,"r"); k=
20     lpav.SetBrush(0,255,0);
21     lpav.SetFontName("Mistral");
22     lpav.SetName(i);
23     lpav.SetClipart( lp.GetPath().."/card/vum
24     lpav.SetOption(LPAV_UNIT,"%");
25     end
26     zz=LPAV_OBJ_FILL_CIRCLE;
27 -- Créé des objets s r

```



lpav.InsertObject

```
result=lpav.InsertObject(type,'name',x,y,'UID');
```

Explication

Result=lpav.InsertObject(type,'name',x,y,'UID'); permet d'insérer dans l'ActiveView un nouvel objet ayant un rapport avec un objet de la directory, qui deviendra l'objet par défaut.

Paramètres en entrée

type le type d'objet à insérer, la liste est fourni dans le fichier loriotinit.lua en fonction du type choisi LoriotPro intégrera automatiquement l'iconographie associé.

Voir le tableau de la fonction lpav.Insert.

'name' le nom de l'objet

x la position x dans l'ActiveView.

Y la position y dans l'ActiveView.

'UID' l'UID d'un objet de la directory associé a l'objet de l'ActiveView. En fonction de l'UID et du type d'objet de directory LoriotPro peut insérer des options spécifiques.

Paramètres en sortie

Result = 1 si l'objet est inséré sinon nil

Exemple

Cet exemple est complexe, il permet d'importer l'internetwork map dans une ActiveView, il servira d'exemple d'utilisation de l'api lpav.

```
dofile(lp.GetPath().."/config/script/loriotinit.lua");
lpav.DeleteAllSelected();

--lpav.InsertObject(LPAV_OBJ_ORGANIZATION,"toiptest",10,10,4784914268657025847);
--lpav.InsertObject(LPAV_OBJ_HOST_1,"name",20,40,4784914431865783096);
lpav.SetGlobalOption(LPAV_G_MAX_X,3000);
lpav.SetGlobalOption(LPAV_G_MAX_Y,3000);
lpav.SetGlobalOption(LPAV_G_ZOOMAUTO,0);

net,mask=lp.GetFirstNetwork();
i=1;
if i==1 then
while net~=nil do
```

```

        Ip.Trace(net..mask);
        if (Ip.GetNetworkInformation(net,mask,"a")) then
            --Ip.Trace(a.name..net..mask);
            --if (a.nb_router>0) then
Ipav.InsertObject(LPAV_OBJ_NETWORK,a.name,a.x,a.y,a.UID) end
            if (a.nb_host>0) then Ipav.InsertObject(LPAV_OBJ_NETWORK,net,a.x,a.y,a.UID)
end
                Ipav.InsertObject(LPAV_OBJ_NETWORK,net,a.x,a.y,a.UID);
            end
            net,mask=Ip.GetNextNetwork(net,mask);
        end
    end
end

Ipav.RefreshMap();

router=Ip.GetFirstRouter();
while router~=nil do
--Ip.Trace("router >"..net);
if Ip.GetRouterInformation(router,"a") ==1 then
    --Ip.Trace("router >"..a.name..router.." "..a.x.." "..a.y);
    Ipav.InsertObject(LPAV_OBJ_ROUTER,a.name, a.x,a.y,a.UID);
    Ipav.GetInformation("b1");
    ip=Ip.GetFirstIP();
        while ip~=nil do
            Ip.GetIPInformation(ip,"aa");
            if aa.grouppref==Ip.atoip(router) then
                --Ip.Trace("Found router");
                net,mask=Ip.GetNetworkFromIP(ip);
                if net~=nil then
                    --Ip.Trace("Found network");
                    Ip.GetNetworkInformation(net,mask,"aaa");
                    --Ip.Trace("pass GetNetworkInformation");
                    co=Ipav.FindFromAssociatedUID(aaa.UID);
                    if co~=nil then
                        Ipav.GetInformation("b2");
                        --Ip.Trace("pass GetInformation");
                    end
                end
            end
            c3=Ipav.InsertObject(LPAV_OBJ_V_LINE,"connector",1,1,"0");
            --Ip.Trace("Insert Connector >"..b2.UID.." "..b1.UID);
            Ipav.SetOption(LPAV_STATUS ,10);
            Ipav.SetOption(LPAV_OBJECTTYPE ,LPAV_OBJ_CONNECTOR);
            Ipav.SetOption(LPAV_MAPTYPE ,LPAV_OBJ_CONNECTOR);
            Ipav.SetOption(LPAV_CONNECTOR_AUTO_ALIGN ,"1");
            Ipav.SetOption(LPAV_CONNECTED_TO1 ,b2.UID);
            Ipav.SetOption(LPAV_CONNECTED_TO2 ,b1.UID);
            --Ip.Trace("SetOption");
        end
    end
    ip=Ip.GetNextIP(ip);
end
end

Ipav.RefreshMap();
router=Ip.GetNextRouter(router);
end
Ipav.RefreshMap();
Ipav.Save(Ip.GetPath().."/config/active-view/v.cmp");
--Ipav.ClearAllSelected();
--Ipav.DeleteAll();

```

LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

The screenshot displays the LoriotPro Script Editor interface. The main window shows a Lua script for inserting and managing network objects. The script includes comments for object insertion and logic for iterating through networks and routers.

```
10 dofile(lp.GetPath().."/config/script/loriotinit.lua");
11 lpav.DeleteAllSelected();
12
13 --lpav.InsertObject(LPAV_OBJ_ORGANIZATION,"toiptest",10,10,4784914268657025847);
14 --lpav.InsertObject(LPAV_OBJ_HOST_1,"name",20,40,4784914431865783096);
15 lpav.SetGlobalOption(LPAV_G_MAX_X,3000);
16 lpav.SetGlobalOption(LPAV_G_MAX_Y,3000);
17 lpav.SetGlobalOption(LPAV_G_ZOOMAUTO,0);
18
19 net,mask=lp.GetNetworkFirst();
20 i=1;
21 -if i==1 then
22 -while net~=nil do
23   lp.Trace(net..mask);
24   if (lp.GetNetworkInformations(net,mask,"a") the
25     --lp.Trace(a.name..net..mask);
26     --if (a.nb_router>0) then lpav.InsertObject(LPA
27     if (a.nb_host>0) then lpav.InsertObject(LPA
28     lpav.InsertObject(LPAV_OBJ_NETWORK,net,a.
29   end
30   net,mask=lp.GetNetworkNext(net,mask);
31 end
32 end
33
34 lpav.RefreshMap();
35
36 router=lp.GetRouterFirst();
37 -while router~=nil do
```

The ActiveView Simulator Window shows a network diagram with a central blue router icon and several host icons connected by lines. The diagram is enclosed in a green dashed box.

The Trace Window displays a table of values:

Value Name	Value in	Value out
lp_host	192.168.1.1	192.168.1.
lp_index		
lp_trace	1	1
lp_old		
lp_buffer		error
lp_value		0

Below the table, the Trace Window shows a list of IP addresses:

```
[82.65.181.0255.255.255.0]
[192.168.1.0255.255.255.0]
```

Ipav.GetInformation

```
result=Ipav.GetInformation(« array »);
```

Explication

La fonction `result=Ipav.GetInformation(« array »)`; retourne un tableau avec des informations sur l'objet par défaut.

Paramètres en entrée

« array » le nom du tableau pour retourner les paramètres de l'objet par défaut.

Paramètres en sortie

Result=1 si le tableau est rempli sinon nil

Variables en retour	description
array['UID']	L'UID sous la forme d'une chaîne de caractère associé à l'objet. Attention, il ne s'agit pas d'un UID de directory mais de l'UID de l'objet par défaut.
array['name']	Le nom de l'objet.
array['fontname']	Le nom de la fonte associée à l'objet
array['x']	La position x de l'objet
array['y']	La position y de l'objet
array['x1']	La largeur de l'objet
array['y1']	La hauteur de l'objet
array['polling']	1 si l'objet réalise un polling
array['polling interval']	L'intervalle de polling si l'objet réalise une action
array['lastpolling']	Le timestamp du dernier polling
array['lastpollingresp']	Le timestamp de la dernière réponse de polling
array['result']	Le dernier résultat du polling
array['expression']	L'expression liée au polling
array['clipartname']	Le non du fichier de clipart associé a l'objet
array['linewidth']	La largeur de ligne de l'objet
array['objtype']	Le type d'objet (voir le fichier loriotinit.lua)
array['maptype']	Le type de mapping associé à l'objet
array['ref1']	La chaîne de caractère Numéro 1 associé à l'objet
array['ref2']	La chaîne de caractère Numéro 2 associé à l'objet
array['ref3']	La chaîne de caractère Numéro 3 associé à l'objet
array['ref4']	La chaîne de caractère Numéro 4 associé à l'objet

Exemple

```
Ip_value = 0;
```

```

lp_buffer ="error";

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

lpav.Load(lp.GetPath().."/config/active-view/sample/cisco3500.cmp");

function displayobjectInformation (co)

if lpav.GetInformation("a") then
lp.Trace("name>"..a.name);
lp.Trace("Object Type ----->");
if a.objtype==LPAV_OBJ_WORLD then lp.Trace("directory world (emf)")
elseif a.objtype==LPAV_OBJ_COUNTRY then lp.Trace("directory country (emf)")
elseif a.objtype==LPAV_OBJ_ORGANIZATION then lp.Trace("directory organization (emf)")
elseif a.objtype==LPAV_OBJ_ORGANIZATION_UNIT then lp.Trace("directory organization unit
(emf)")
elseif a.objtype==LPAV_OBJ_NETWORK then lp.Trace("directory network (emf)")
elseif a.objtype==LPAV_OBJ_HOST_1 then lp.Trace("directory host 1 (emf)")
elseif a.objtype==LPAV_OBJ_ROUTER then lp.Trace("directory router (emf)")
elseif a.objtype==LPAV_OBJ_SHORTCUT_1 then lp.Trace("directory shortcut 1(emf)")
elseif a.objtype==LPAV_OBJ_SHORTCUT_2 then lp.Trace("directory shortcut 2(emf)")
elseif a.objtype==LPAV_OBJ_MRTG then lp.Trace("directory mrtg (emf)")
elseif a.objtype==LPAV_OBJ_PLUGIN then lp.Trace("directory plugin (emf)")
elseif a.objtype==LPAV_OBJ_BADPLUGIN then lp.Trace("directory badplugin (emf)")
elseif a.objtype==LPAV_OBJ_LORIOTPRO then lp.Trace("directory Loriotpro (emf)")
elseif a.objtype==LPAV_OBJ_HOST_2_ROUTER then lp.Trace("directory host 2 router (emf)")
elseif a.objtype==LPAV_OBJ_HOST_3_SWITCH then lp.Trace("directory host 3 switch (emf)")
elseif a.objtype==LPAV_OBJ_HOST_4 then lp.Trace("directory host 4 (emf)")
elseif a.objtype==LPAV_OBJ_HOST_5 then lp.Trace("directory host 5 (emf)")
elseif a.objtype==LPAV_OBJ_HOST_6 then lp.Trace("directory host 6 (emf)")
elseif a.objtype==LPAV_OBJ_VIEWPLUGIN then lp.Trace("directory viewplugin (emf)")
elseif a.objtype==LPAV_OBJ_HOST_SERVER then lp.Trace("directory host server (emf)")
elseif a.objtype==LPAV_OBJ_HOST_IMPRIMANTE then lp.Trace("directory host imprimante
(emf)")
elseif a.objtype==LPAV_OBJ_HOST_PHONE then lp.Trace("directory host phone (emf)")
elseif a.objtype==LPAV_OBJ_RECT then lp.Trace("rectangle")
elseif a.objtype==LPAV_OBJ_CIRCLE then lp.Trace("circle")
elseif a.objtype==LPAV_OBJ_FILL_RECT then lp.Trace("fillrectangle")
elseif a.objtype==LPAV_OBJ_FILL_CIRCLE then lp.Trace("fill circle")
elseif a.objtype==LPAV_OBJ_H_LINE then lp.Trace("h line")
elseif a.objtype==LPAV_OBJ_V_LINE then lp.Trace("v line")
elseif a.objtype==LPAV_OBJ_DIR_EMF then lp.Trace("directory object (emf)")
elseif a.objtype==LPAV_OBJ_TEXT then lp.Trace("texte")
elseif a.objtype==LPAV_OBJ_FILL_TEXT then lp.Trace("fill texte")
elseif a.objtype==LPAV_OBJ_FULL_RECT then lp.Trace("full rectangle")
elseif a.objtype==LPAV_OBJ_FULL_CIRCLE then lp.Trace("full circle")
elseif a.objtype==LPAV_OBJ_H45_LINE then lp.Trace("H line 45")
elseif a.objtype==LPAV_OBJ_V45_LINE then lp.Trace("V line 45")
elseif a.objtype==LPAV_OBJ_FULL_TEXT then lp.Trace("full texte")
elseif a.objtype==LPAV_OBJ_BASIC_SHAPE then lp.Trace("basic shape (emf)")
elseif a.objtype==LPAV_OBJ_ANALOGIC_METER_01 then lp.Trace("analogic meter (emf)")
elseif a.objtype==LPAV_OBJ_HOST_HUB then lp.Trace("host hub")
elseif a.objtype==LPAV_OBJ_CONNECTOR then lp.Trace("connector")
end
lp.Trace("fontname>"..a.fontname);
lp.Trace("x>"..a.x);
lp.Trace("y>"..a.y);
lp.Trace("x1>"..a.x1);
lp.Trace("y1>"..a.y1);

```

```
lp.Trace("polling_interval>"..a.polling_interval);
lp.Trace("lastpolling>"..a.lastpolling);
lp.Trace("lastpollingresp>"..a.lastpollingresp);
lp.Trace("polling>"..a.polling);
lp.Trace("result>"..a.result);
lp.Trace("clipartname>"..a.clipartname);
lp.Trace("linewidth>"..a.linewidth);
lp.Trace("objtype>"..a.objtype);
lp.Trace("maptype>"..a.maptype);
lp.Trace("ref1>"..a.ref1);
lp.Trace("ref2>"..a.ref2);
lp.Trace("ref3>"..a.ref3);
lp.Trace("ref4>"..a.ref4);
lp.Trace("UID>"..a.UID);
return 1;
end
return 0;
end

if lpav.SelectFirst() then
co=lpav.SelectFirst();
  while co~=nil do
    lp.Trace("<<<object>>>");
    displayobjectInformation (co);
    co=lpav.SelectNext();
  end
end

lp.Trace("fin");
```

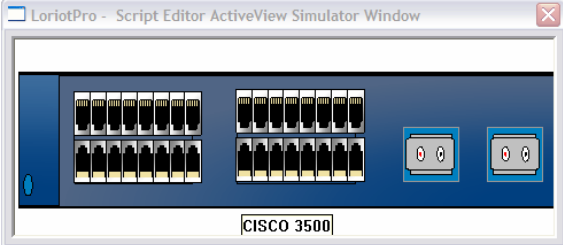
LoriotPro - Script Editor : D:\v4-net\bin\config\script\lpav.GetInformations.lua

```

53 elseif a.objtype==LPAV_OBJ_CONNECTOR then lp.Trace("connector")
54 end
55 lp.Trace("fontname>".a.fontname);
56 lp.Trace("x>".a.x);
57 lp.Trace("y>".a.y);
58 lp.Trace("x1>".a.x1);
59 lp.Trace("y1>".a.y1);
60 lp.Trace("polling_interval>".a.polling_interval);
61 lp.Trace("lastpolling>".a.lastpolling);
62 lp.Trace("lastpollingresp>".a.lastpollingresp);
63 lp.Trace("polling>".a.polling);
64 lp.Trace("result>".a.result);
65 lp.Trace("clipartname>".a.clipartname);
66 lp.Trace("linewidth>".a.linewidth);
67 lp.Trace("objtype>".a.objtype);
68 lp.Trace("maptype>".a.maptype);
69 lp.Trace("ref1>".a.ref1);
70 lp.Trace("ref2>".a.ref2);
71 lp.Trace("ref3>".a.ref3);
72 lp.Trace("ref4>".a.ref4);
73 lp.Trace("UID>".a.UID);
74 return
75 end
76 return
77 end
80 -if Inav.

```

LoriotPro - Script Editor ActiveView Simulator Window



LoriotPro - Script Editor Trace Window

```

[<<<object>>]
[name>CISCO 3500]
[Object Type ----->]
[directory object (emf)]
[fontname>System]
[x>2]
[y>13]
[x1>652]
[y1>66]
[polling_interval>15]
[lastpolling>0]
[lastpollingresp>0]
[polling>0]
[result>(null)]
[clipartname>card/devices/cisco/ciscocatalyst3500.emf]
[linewidth>1]
[objtype>36]
[maptype>2]
[ref1>(null)]
[ref2>(null)]
[ref3>(null)]
[ref4>(null)]
[UID>4604358336068976842]
[<<<object>>]
[name>Connectors]
[Object Type ----->]

```

Value Name	Value
lp_host	192
lp_index	
lp_trace	1
lp_oid	
lp_buffer	
lp_value	

Ipav.SetPosition

```
result=Ipav.SetPosition(x,y);  
result=Ipav.SetPosition(x,y,x1,y1);
```

Explication

Positionne en x,y l'objet par défaut, si x1 y1 modifie la taille de l'objet.

Paramètres en entrées

X la position x de l'objet
Y la position y de l'objet
X1 la largeur de l'objet
Y1 la hauteur de l'objet

Paramètres en sortie

Result = 1 sinon nil

Exemple

```
Ip.Trace("start");  
dofile(lp.GetPath().."/config/script/loriotinit.lua");  
  Ip_value = 0;  
  Ip_buffer ="error";  
Ipav.Load(lp.GetPath().."/config/active-view/sample/cisco3500.cmp");  
  
for i=1,5 do  
co=Ipav.SelectFirst();  
  while co~=nil do  
    if Ipav.GetInformation("a") then  
      if a.objtype~=LPAV_OBJ_CONNECTOR then  
        x,y,x1,y1=Ipav.GetPosition();  
        if x>1000 then x=10 end  
        if y>1000 then y=10 end  
        Ipav.SetPosition(x+100,y+100,x1,y1);  
      end  
    end  
    co=Ipav.SelectNext();  
  end  
end  
Ipav.RefreshMap();  
end  
  
Ipav.RefreshMap();  
Ip.Trace("fin");
```


LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

The screenshot displays the LoriotPro Script Editor interface. The main window shows a Lua script for managing devices. The script includes a trace at the start, a loop to process 5 devices, and a trace at the end. Two auxiliary windows are open: 'LoriotPro - Script Editor ActiveView Simulator Window' showing a network map with a 'CISCO 3500' device highlighted, and 'LoriotPro - Script Editor Trace Window' showing the execution flow from '[start]' to '[fin]'. A data table is visible at the bottom of the editor.

```
1 lp.Trace("start");
2 dofile(lp.GetPath().."/config/script/loriotinit.lua");
3 lp_value = 0;
4 lp_buffer = "error";
5 lpav.Load(lp.GetPath().."/config/active-view/sample/cisco3500.cmp");
6
7
8 -for i=1,5 do
9 co=lpav.SelectFirst();
10 - while co~=nil do
11 - if lpav.GetInformations("a") then
12 - if a.objtype~=LPAV_OBJ_CONNECTOR then
13 x,y,x1,y1=lpav.GetPosition();
14 if x>1000 then x=10 end
15 if y>1000 then y=10 end
16 lpav.SetPosition(x+100,y+100,x1,y1
17 end
18 end
19 co=lpav.SelectNext();
20 end
21 lpav.RefreshMap();
22 end
23
24 lpav.RefreshMap();
25 lp.Trace("fin");
```

Value Name	Value in	Value out
lp_host	192.168.1.1	192.168.1.1
lp_index		
lp_trace	1	1

Ipav.SetSize

```
result=Ipav.SetSize(x,y);
```

Explication

Modifie la taille de l'objet.

Paramètres en entrées

X1 la largeur de l'objet
Y1 la hauteur de l'objet

Paramètres en sortie

Result = 1 sinon nil

Ipav.GetPosition

x,y,x1,y1=Ipav.GetPosition() ;

Explication

Fournit les coordonnées de l'objet par défaut.

Paramètres en entrées

L'objet par défaut

Paramètres en sortie

X la position x de l'objet
Y la position y de l'objet
X1 la largeur de l'objet
Y1 la hauteur de l'objet

Sinon nil

Ipav.SetCoord

Result=Lpav.SetCoord(x,y);

Explication

Positionne l'objet par défaut en le centrant sur x,y.

Paramètres en entrées

L'objet par défaut

X la position x (centré) de l'objet

Y la position y (centré) de l'objet

Paramètres en sortie

Result=1 Sinon nil

Ipav.RefreshMap

Ipav.RefreshMap();

Explication

Force le rafraichissement « refresh » graphique de l'ActiveView. Cette option permet de forcer un rafraichissement de l'ActiveView si le script modifie les graphiques sans avoir besoin d'attendre le rafraichissement standard.

Attention

Cette fonction est très lente. Pour 600 scripts utilisant 1 fois chacun RefreshMap sur une ActiveView il faut environs 180s pour faire le cycle de polling contre environs 15s.

Ipav.SetClipart

Ipav.SetClipart(« file.emf »)

Explication

Change ou charge le clipart associé à l'objet par défaut.

Attention : pas supportée pour les objets suivant :

```
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_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_CONNECTOR = 144;
```

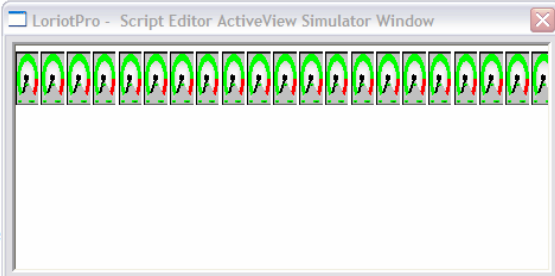
Paramètres en entrée

« file.emf » le nom du fichier a chargé.

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
  
dofile(Ip.GetPath().."/config/script/loriotinit.lua");  
  
zz=LPAV_OBJ_ANALOGIC_METER_01;  
  
-- Créé des objects r s  
k=0;  
for i=0,20 do  
    Ipav.Insert(10*i,1,9,9,zz);  
    if (k==0) then Ipav.SetRef(1,"r"); k=1; else Ipav.SetRef(1,"s"); k=0; end  
    Ipav.SetBrush(0,255,0);  
    Ipav.SetFontName("Mistral");  
    Ipav.SetName(i);  
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");  
Ipav.SetOption(LPAV_UNIT,"%");  
end  
Ipav.RefreshMap();
```

```
LoriotPro - Script Editor : D:\v4-net\bin\config\script\lpav.insert.lua
Files Edit View Insert Search Options Compiler Value
1  -- Loriotpro V4
2  -- Ce fichier doit impérativement se trouver dans bin/config/script
3  -- Parametres passés au script
4  -- lp_index l'index a ajouté au requete ".1"
5  -- lp_oid l'oid snmp passé ou une valeur "ifnumber"
6  -- lp_host l'add ip du host en string "127.0.0.1"
7  -- valeurs à retourner impérativement
8  lp_value = 0;
9  lp_buffer = "error";
10
11 dofile(lp.GetPath().."/config/script/loriotinit.lua");
12
13 zz=LPAV_OBJ_ANALOGIC_METER_01;
14
15 -- Créé des objects r s
16 k=0;
17 --for i=0,20 do
18     lpav.Insert(10*i,1,9,9,zz);
19     if (k==0) then lpav.SetRef(1,"r"); k=1; else
20     lpav.SetBrush(0,255,0);
21     lpav.SetFontName("Mistral");
22     lpav.SetName(i);
23     lpav.SetClipart(lp.GetPath().."/card/vumeters/vumeter8.emf");
24     lpav.SetOption(LPAV_UNIT,"%");
25 end
26 lpav.RefreshMap();
27
```



Ipav.SetToBackGround

```
Ipav.SetToBackGround();
```

Explication

Place l'objet par défaut en arrière plan.

Ipav.SetToForGround

```
Ipav.SetToForGround();
```

Explication

Place l'objet par défaut en avant plan.

Ipav.SetDown

```
Ipav.SetDown();
```

Explication

Tous les objets de l'ActiveView possède une logique de couche (plan). Cette fonction place l'objet par défaut un plan plus bas.

Ipav.Setup

```
Ipav.Setup();
```

Explication

Tous les objets de l'ActiveView possèdent une logique de couche (plan). Cette fonction place l'objet par défaut un plan plus en avant.

Ipav.SetName

```
result=Ipav.SetName(« name »);
```

Explication

Change le nom de l'objet par défaut.

Paramètres en entrée

« name » le nom de l'objet

Paramètres en sortie

Result=1 sinon nil

Ipav.SetBrush

```
result=Ipav.SetBrush(r,g,b);
```

Explication

Change la couleur de fond de l'objet par défaut.
Attention le pinceau « brush » des objets dynamique est géré dynamiquement en fonction des filtres de résultats appliqués.

Paramètres en entrée

r la composante rouge (0 – 255)
g la composante verte (0 – 255)
b la composante bleue (0 – 255)

Paramètres en sortie

Result=1 sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer = "error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_FILL_RECT;

-- Créé des objects r s
k=0;
for i=0,20 do
    Ipav.Insert(10*i,1,9,9,zz);
    if (k==0) then Ipav.SetRef(1,"r"); k=1; else Ipav.SetRef(1,"s"); k=0; end
    Ipav.SetBrush(0,i*5,i*10);
    Ipav.SetFontName("courrier");
    Ipav.SetName(i);
end

Ip.Trace("fin");
Ipav.RefreshMap();
```

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

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  dofile(lp.GetPath().."/config/script/loriotinit.lua");
6
7  zz=LPAV_OBJ_FILL_RECT;
8
9  -- Créé des objects r s
10 k=0;
11 -- for i=0,20 do
12   lpav.Insert(10*i,1,9,9,zz);
13   if (k==0) then lpav.SetRef(1,"r"); k=1; else lpav.SetRef(1,"s"); k=0; end
14   lpav.SetBrush(0,i*5,i*10);
15   lpav.SetFontName("courrier");
16   lpav.SetName(i);
17 end
18
19 lp.Trace("fin");
20 lpav.RefreshMap();
21
```

An "ActiveView Simulator Window" is overlaid on the script, showing a grid of 20 columns. The first 10 columns are black, and the remaining 10 columns are blue. Below the simulator window, a table displays the following data:

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65

Ipav.SetFontName

```
result=Ipav.SetFontName("font name");
```

Explication

Change la fonte de caractère utilisée par l'objet par défaut. La fonte doit exister et être chargée pour pouvoir être utilisée.

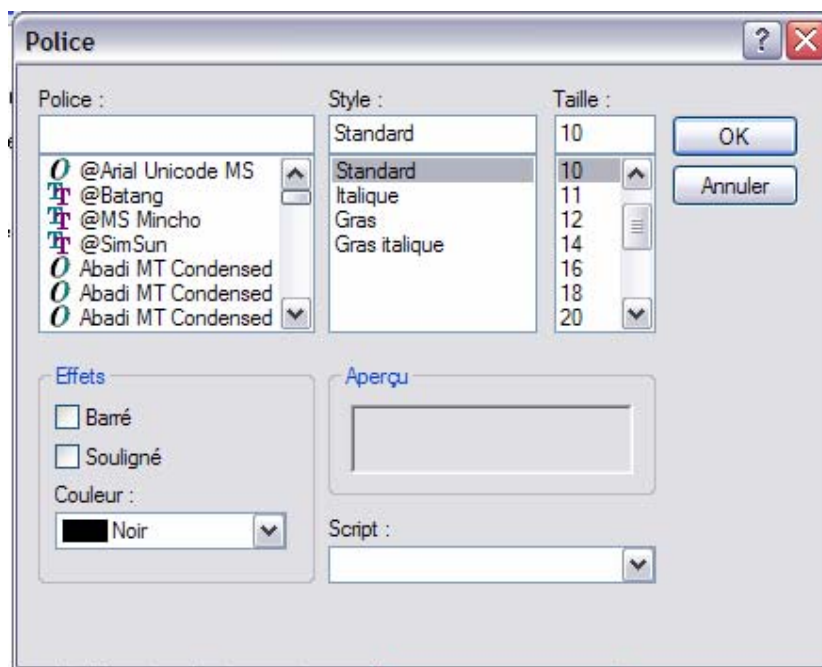
Paramètres en entrée

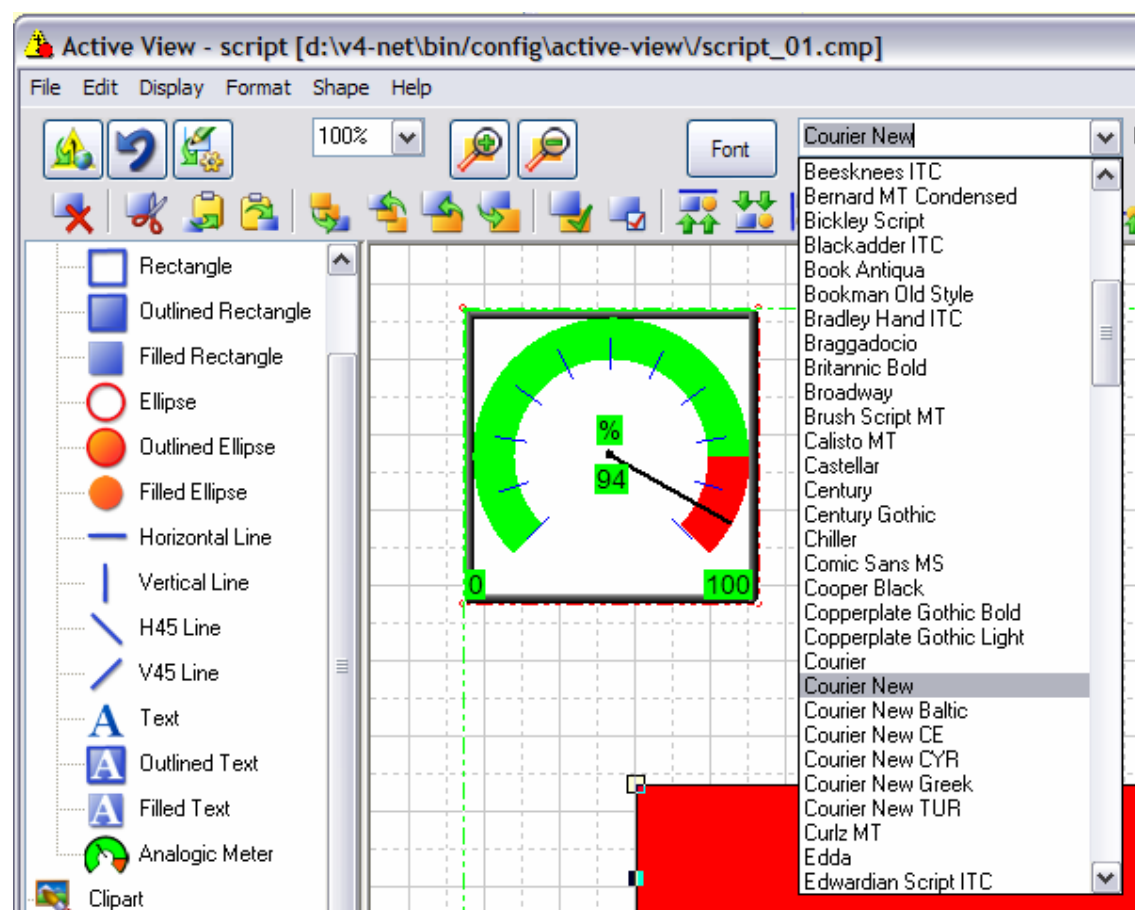
"font name" le nom de la fonte.

Paramètres en sortie

Result=1 sinon nil

Exemple





```

lp_value = 0;
lp_buffer = "error";

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

zz=LPAV_OBJ_FILL_RECT;

-- Créé des objects r s
k=0;
for i=0,20 do
    lpav.Insert(10*i,1,9,9,zz);
    lpav.SetFontName("courier");
end

lp.Trace("fin");
lpav.RefreshMap();
    
```


Ipav.SetLineColor

```
result=Ipav.SetLineColor(r,g,b);
```

Explication

Change la couleur de la ligne de l'objet par défaut.
Attention certains objets n'ont pas de ligne.

Paramètres en entrée

r la composante rouge (0 – 255)
g la composante verte (0 – 255)
b la composante bleue (0 – 255)

Paramètres en sortie

Result=1 sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer = "error";

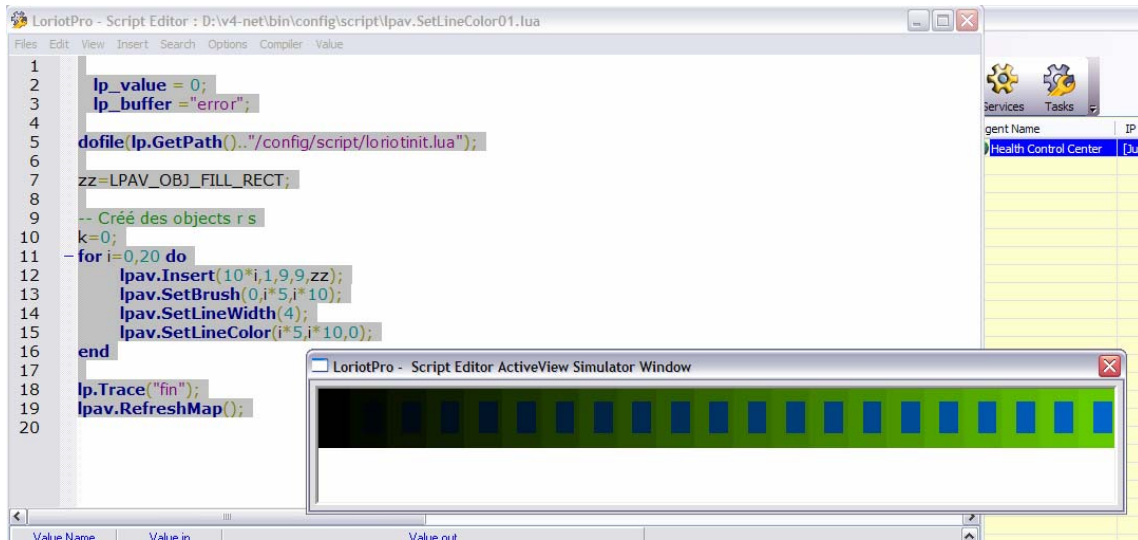
dofile(lp.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_FILL_RECT;

-- Créé des objects r s
k=0;
for i=0,20 do
    Ipav.Insert(10*i,1,9,9,zz);
    Ipav.SetBrush(0,i*5,i*10);
    Ipav.SetLineWidth(4);
    Ipav.SetLineColor(i*5,i*10,0);
end

lp.Trace("fin");
Ipav.RefreshMap();
```

LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA



Ipav.SetLineWidth

```
result=Ipav.SetLineWidth(l);
```

Explication

Change l'épaisseur de la ligne utilisée pour l'objet par défaut.
Attention certains objets n'ont pas de ligne.

Paramètres en entrée

l l'épaisseur de la ligne

Paramètres en sortie

Result=1 sinon nil

Exemple

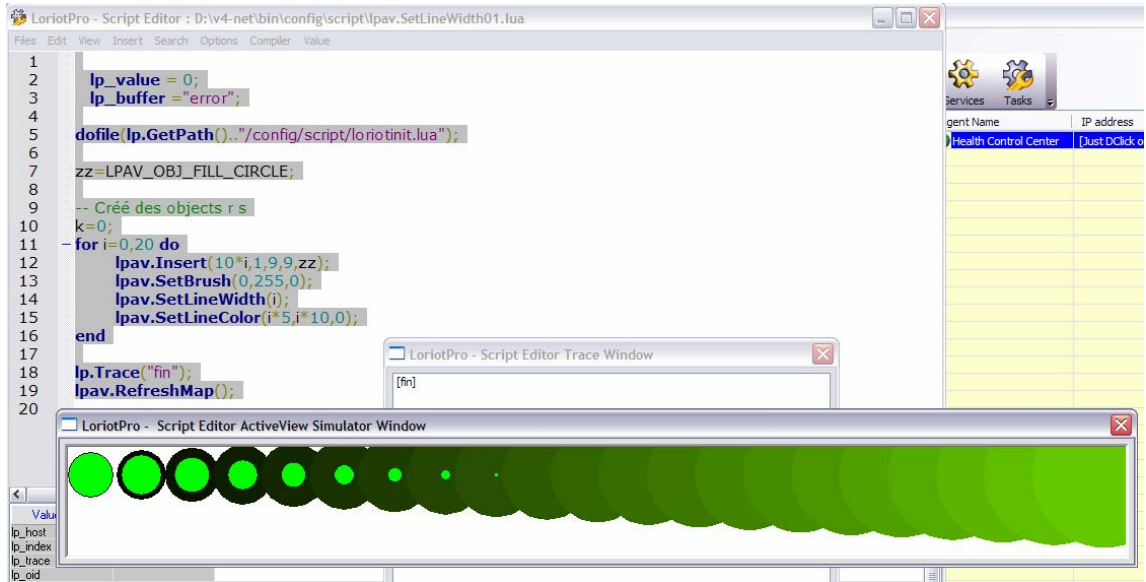
```
Ip_value = 0;
Ip_buffer = "error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_FILL_CIRCLE;

-- Créé des objects r s
k=0;
for i=0,20 do
    Ipav.Insert(10*i,1,9,9,zz);
    Ipav.SetBrush(0,255,0);
    Ipav.SetLineWidth(i);
    Ipav.SetLineColor(i*5,i*10,0);
end

Ip.Trace("fin");
Ipav.RefreshMap();
```



Ipav.SetOption

```
Ipav.SetOption(optionref, 'value');
```

Explication

Cette fonction permet de modifier les paramètres de l'objet par défaut.

Paramètres en entrée

Optionref le paramètre à modifier. La liste des paramètres disponibles est incluse dans le fichier loriotinit.lua

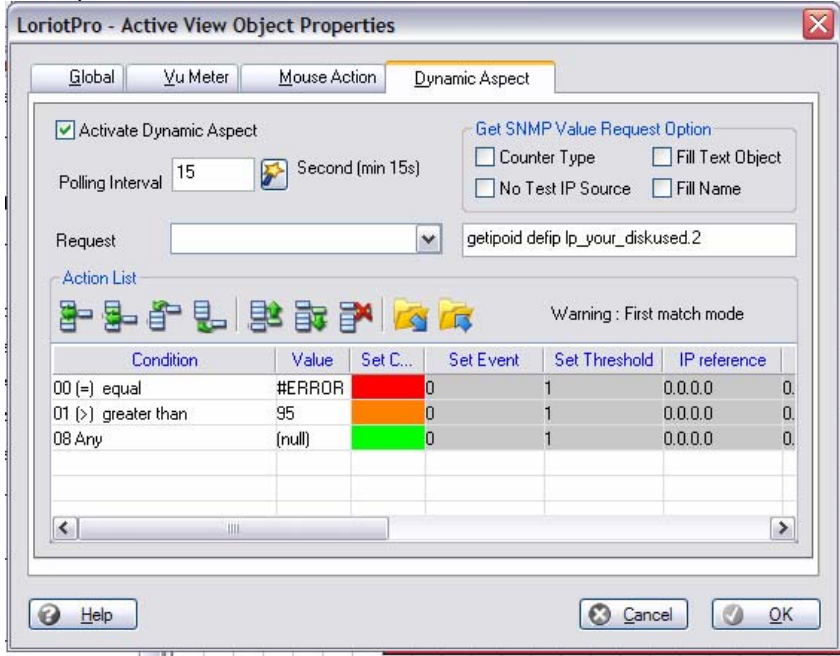
```
dofile(lp.GetPath().."/config/script/loriotinit.lua");
```

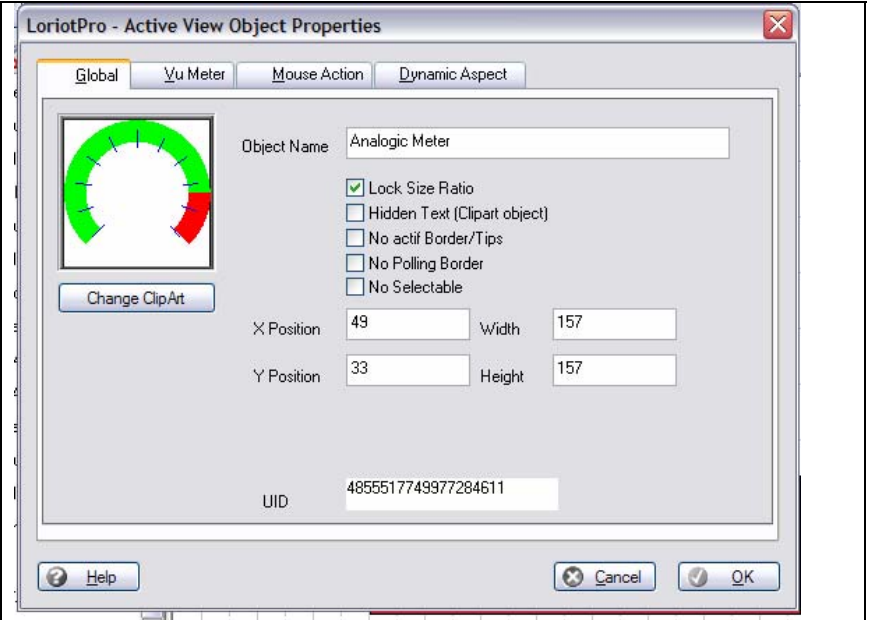
« value » la valeur du nouveau paramètre

Optionref	Value
LPAV_NAME	« name » Le nom de l'objet
LPAV_FONTNAME	« fonte » Le nom de la fonte utilisée
LPAV_FONTSIZE	« size » La 'taille' de la fonte
LPAV_LINEWIDTH	« largeur » La largeur de la fonte
LPAV_LINESTYLE	« style » Le style de la fonte
LPAV_EXPRESSION	« expression » L'expression utilisé par le scheduler pour exécuter la requete
LPAV_POLLING	« 1 » oui « 0 » non l'expression est executée à intervalle régulier 0 l'exécution est arrêté.
LPAV_POLLING_INTERVAL	« intervalle » L'interval de l'exécution de la requête (mini 15 s)
LPAV_SELECTED	« 1 » oui « 0 » non Cet objet est selectionné
LPAV_OBJECTTYPE	Le type de l'objet :
LPAV_HIDDEN_TEXT	« 1 » hidden « 0 » no hidden le texte associé à l'objet est affiché ou non
LPAV_NOBORDER	« 1 » noborder « 0 » border l'objet possède un bord lorsque la souris passe au dessus
LPAV_TEXTMODE	
LPAV_TEXTSTYLE	
LPAV_UNIT	« unit » L'unité utilisé par les Vumeter
LPAV_X	« x » La position x de l'objet
LPAV_X1	« x1 » la largeur de l'objet
LPAV_Y	« y » La position y de l'objet
LPAV_Y1	« y1 » La hauteur de l'objet
LPAV_TYPE	« type » ATTENTION au incoherence possible LPAV_OBJ_WORLD =0; LPAV_OBJ_COUNTRY =1;

	<pre> 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; LPAV_OBJ_HOST_HUB = 46; LPAV_OBJ_CONNECTOR =144; </pre>
LPAV_CONNECTED_TO1	<pre> « uid » L'UID de l'objet connecté </pre>
LPAV_CONNECTED_TO2	<pre> « uid » L'UID de l'objet connecté lp_value = 0; lp_buffer ="error"; dofile(lp.GetPath().."/config/script/loriotinit.lua"); zz=LPAV_OBJ_ANALOGIC_METER_01; -- Créé des objects r s k=0; for i=0,20 do lpav.Insert(10*i,1,9,9,zz); if (k==0) then lpav.SetRef(1,"r"); k=1; else lpav.SetRef(1,"s"); k=0; end lpav.SetBrush(0,255,0); lpav.SetFontName("Mistral"); </pre>

	<pre> lpav.SetName(i); lpav.SetClipart(lp.GetPath().."/card/vumeters/vumeter8.emf"); lpav.SetOption(LPAV_UNIT,"%"); end zz=LPAV_OBJ_FILL_CIRCLE; -- Créé des objects s r for i=0,20 do lpav.Insert(10*i,30,9,9,zz); if (k==0) then lpav.SetRef(1,"s"); k=1; else lpav.SetRef(1,"r"); k=0; end lpav.SetBrush(0,255,0); lpav.SetName("test-r"..i); lpav.SetOption(LPAV_LINESTYLE,55); end lpav.ClearAllSelected() --lpav.DeleteAllSelected(); i=lpav.FindRef(1,0,"r","a"); if (i>0) then for j=0, i do -- lp.Trace("pass"..j); if lpav.SelectUID(a[j]) then lpav.SetLineColor(255,45,89); lpav.SetLineWidth(3); end end end i=lpav.FindRef(1,0,"s","aa"); if (i>0) then for j=0, i do --lp.Trace("pass"..j); if lpav.SelectUID(aa[j]) then lpav.SetLineColor(255,45,255); lpav.SetLineWidth(5); x,y,x1,y1=lpav.GetPosition(); lpav.SetPosition(x,y+10,x1,y1); lpav.Insert(1,1,100,100,LPAV_OBJ_CONNECTOR); lpav.SetOption(LPAV_CONNECTED_TO1,a[j]); lpav.SetOption(LPAV_CONNECTED_TO2,aa[j]); lpav.SetOption(LPAV_LINEWIDTH,1); lpav.SetLineColor(100,45,45); end end end lp.Trace("fin"); lpav.RefreshMap(); </pre>
LPAV_ISRATIO	« 1 » isratio « 0 » non les poignés de redimensionnement de l'objet respectent la proportion.
LPAV_LEFT_MOUSE_MENU	« 1 » oui « 0 » non Le menu contextuel associé a l'objet s'ouvre avec le bouton gauche de la souris.
LPAV_RESULT_IS_TEXT	« 1 » oui « 0 » non Le resultat de l'expression remplace le texte de l'objet de type texte LPAV_OBJ_TEXT = 37; LPAV_OBJ_FILL_TEXT = 38; LPAV_OBJ_FULL_TEXT = 43;
LPAV_RESULT_IS_NAME	« 1 » oui « 0 » non

	Le résultat de l'expression remplace le nom de l'objet.
LPAV_REF1	« texte » permet de définir la variable ref1
LPAV_REF2	« texte » permet de définir la variable ref2
LPAV_REF3	« texte » permet de définir la variable ref3
LPAV_REF4	« texte » permet de définir la variable ref4
LPAV_ISCOUNTER	« 1 » oui « 0 » non utilisé pour les vumeter 
LPAV_CONNECTOR_AUTO_ALIGN	« 1 » oui « 0 » non le connecteur (lignes entre objets) se positionne automatiquement. Attention ceci n'est valable que pour l'objet de type connector : LPAV_OBJ_CONNECTOR =144;
LPAV_NOIPSOURCE_TEST	« 1 » oui « 0 » non Ne teste pas l'adresse IP fourni par le paquet de la réponse à l'expression.
LPAV_NOSELECTABLE	« 1 » oui « 0 » non

	
LPAV_CLIPART	« filename » permet de changer le clipart associé à un objet supportant les cliparts.
LPAV_STATUS	
LPAV_MAPTYPE	

Paramètres en sortie

Result=1 sinon nil

Exemple

```

lp_value = 0;
lp_buffer ="error";

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,20 do
    lpav.Insert(10*i,1,9,9,zz);
    lpav.SetOption(LPAV_X,10*i);
    lpav.SetOption(LPAV_Y,1);
    lpav.SetOption(LPAV_X1,10*i);
    lpav.SetOption(LPAV_Y1,10*i);
    if (k==0) then lpav.SetRef(1,"r"); k=1; else lpav.SetRef(1,"s"); k=0; end
    lpav.SetBrush(0,255,0);
    lpav.SetOption( LPAV_FONTNAME,"Mistral");
    lpav.SetOption(LPAV_NAME,"test");
    lpav.SetOption( LPAV_CLIPART , lp.GetPath().."/card/vumeters/vumeter8.emf");
    lpav.SetOption(LPAV_UNIT,"%");
end
zz=LPAV_OBJ_FILL_CIRCLE;
-- Créé des objects s r
for i=0,20 do
    lpav.Insert(10*i,30,9,9,zz);

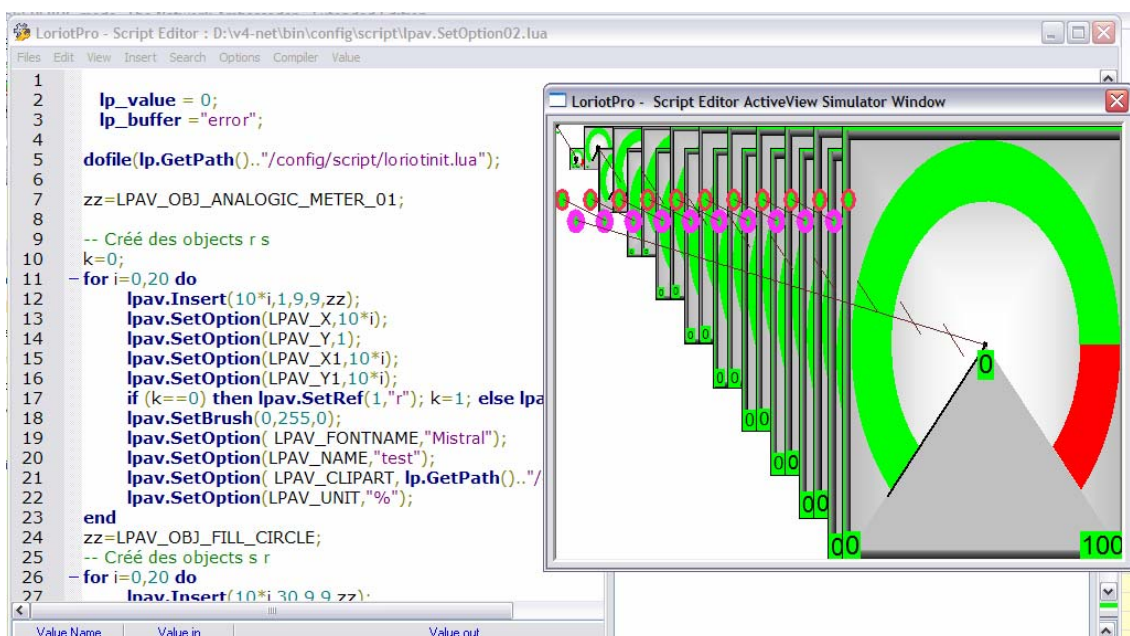
```

```

        if (k==0) then lpav.SetRef(1,"s"); k=1; else lpav.SetRef(1,"r"); k=0; end
        lpav.SetBrush(0,255,0);
        lpav.SetName("test-r"..i);
lpav.SetOption(LPAV_LINestyle,55);
end

lpav.ClearAllSelected()
--lpav.DeleteAllSelected();
i=lpav.FindRef(1,0,"r","a");
if (i>0) then
    for j=0, i do
        -- lp.Trace("pass" ..j);
        if lpav.SelectUID(a[j]) then
            lpav.SetLineColor(255,45,89);
            lpav.SetLineWidth(3);
        end
    end
end
end
i=lpav.FindRef(1,0,"s","aa");
if (i>0) then
    for j=0, i do
        --lp.Trace("pass" ..j);
        if lpav.SelectUID(aa[j]) then
            lpav.SetLineColor(255,45,255);
            lpav.SetLineWidth(5);
            x,y,x1,y1=lpav.GetPosition();
            lpav.SetPosition(x,y+10,x1,y1);
            lpav.Insert(1,1,100,100,LPAV_OBJ_CONNECTOR);
            lpav.SetOption(LPAV_CONNECTED_TO1,a[j]);
            lpav.SetOption(LPAV_CONNECTED_TO2,aa[j]);
            lpav.SetOption(LPAV_LINEWIDTH,1);
            lpav.SetLineColor(100,45,45);
        end
    end
end
end
lp.Trace("fin");
lpav.RefreshMap();

```



Ipav.SelectUID

result=Ipav.SelectUID("uid")

Explication

Change l'objet graphique par défaut en utilisant l'«UID » du nouvel objet graphique.

Paramètres en entrée

« uid » l'UID sous forme de chaîne de caractères de l'objet graphique

Paramètres en sortie

Result=1 alors l'objet graphique a été trouvé et il est maintenant l'objet graphique par défaut.

sinon nil

Exemple

```
i=Ipav.FindRef(1,0,"r","a");
if (i>0) then
  for j=0, i do
    -- Ip.Trace("pass"..j);
    if Ipav.SelectUID(a[j]) then
      Ipav.SetLineColor(255,45,89);
      Ipav.SetLineWidth(3);
    end
  end
end
end
```

Ipav.SetRef

result=Ipav.SetRef(number (1-4), "string")

Explication

Set une variable statique de l'objet graphique. Un objet graphique possède 4 variables de référence. Ces variables sont utilisées pour créer des groupes d'objets graphiques ou pour une usage de référencement d'objets.

Paramètres en entrée

Number le numero de la référence 1 2 3 4
« string » une chaîne de caractères contenant une informations de référencement

Paramètres en sortie

Result=1 egal ok sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer ="error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");
zz=LPAV_OBJ_FILL_CIRCLE;
-- Créé des objects s r
for i=0,20 do
    Ipav.Insert(10*i,30,9,9,zz);
    if (k==0) then Ipav.SetRef(1,"s"); k=1; else Ipav.SetRef(1,"r"); k=0; end
    Ipav.SetBrush(0,255,0);
    Ipav.SetName("test-r"..i);
Ipav.SetOption(LPAV_LINESTYLE,55);

end
i=Ipav.FindRef(1,0,"r","a");
if (i>0) then
    for j=0, i do
--        Ip.Trace("pass"..j);
        if Ipav.SelectUID(a[j]) then
            Ipav.SetLineColor(255,45,89);
            Ipav.SetLineWidth(3);
        end
    end
end
end
Ipav.RefreshMap();
```

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

```
8  lpav.Insert(10*i,30,9,9,zz);
9  if (k==0) then lpav.SetRef(1,"s"); k=1; else lpav.SetRef(1,"r"); k=0; end
10 lpav.SetBrush(0,255,0);
11 lpav.SetName("test-r"..i);
12 lpav.SetOption(LPAV_LINESTYLE,55);
13
14 end
15 i=lpav.FindRef(1,0);
16 -if (i>0) then
17   -for j=0, i do
18     -- lp.Trace("pass"
19     -if lpav.Se
20       lpav
21       lpav
22     end
23   end
24 end
25 lpav.RefreshMap()
26
```

Overlaid on the script is the "LoriotPro - Script Editor ActiveView Simulator Window". This window displays a horizontal row of 15 green circles, each with a red outline, representing a map or data visualization. Below the simulator window, a table shows the current values of variables:

Value Name	Value in	Value
lp_host	82.65.181.65	82
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		err
lp_value		0

Ipav.FindRef

```
“uid”=Ipav.FindRef(number (1-4), mode, “string”);
objet_number=Ipav.FindRef(number (1-4), mode, “string”, “array”);
```

Explication

FindRef permet de faire une sélection multiple d’objets graphiques en utilisant une variable de référence de l’objet graphique. Un objet graphique possède 4 variables de référence. Ces variables sont utilisées pour créer des groupes d’objets graphiques ou pour un usage de référencement d’objets. Si string est égal à la variable de référence alors le premier objet graphique de la liste en mémoire est sélectionné et devient l’objet graphique par défaut. Si la variable array est ajoutée alors les objets graphiques répondant à la référence « string » sont sélectionnés et un tableau est retourné pour pouvoir travailler sur cette liste sélectionnée.

Paramètres en entrée

Number	le numero de la référence 1 2 3 4
Mode	1 nocase sensitive 0 case sensitive
« string »	une chaine de caractères de recherche
« array »	le nom d’un tableau pour stoker les résultats

Paramètres en sortie

```
“uid”=Ipav.FindRef(number (1-4), mode, “string”);
```

l’uid de l’objet graphique par défaut ou nil.

```
objet_number=Ipav.FindRef(number (1-4), mode, “string”, “array”);
```

Le nombre d’objets graphiques dans le tableau ou nil.

Variables en retour	description
arrayname [0]	L’uid de l’objet graphique 0 dans la liste
arrayname [1]	L’uid de l’objet graphique 1 dans la liste.
...	
arrayname [x]	L’uid de l’objet graphique 1 dans la liste.

Exemple

Voir Ipav.SetRef

Ipav.GetRef

```
“string”=Ipav.GetRef(number(1-4));  
objet_number=Ipav.FindRef(number (1-4), mode, “string”,“array”);
```

Explication

GetRef permet de récupérer les références associées à l'objet graphique par défaut. Un objet graphique possède 4 variables de référence. Ces variables sont utilisées pour créer des groupes d'objets graphiques ou pour un usage de référencement d'objets.

Paramètres en entrée

Number le numero de la référence 1 2 3 4

Paramètres en sortie

« string » la chaîne de caractères de la référence ou nil (si il n'y a pas de chaine).

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
  
dofile(Ip.GetPath().."/config/script/loriotinit.lua");  
    Ipav.Insert(10,30,99,99,LPAV_OBJ_FILL_CIRCLE);  
    Ipav.SetRef(1,"ref1");  
    Ipav.SetRef(2,"ref2");  
    Ipav.SetRef(3,"ref3");  
    Ipav.SetRef(4,"ref4");  
  
Ip.Trace(Ipav.GetRef(1));  
Ip.Trace(Ipav.GetRef(2));  
Ip.Trace(Ipav.GetRef(3));  
Ip.Trace(Ipav.GetRef(4));  
  
Ipav.RefreshMap();  
Ip_buffer = "ok";
```


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

```
1  lp_value = 0;
2  lp_buffer = "error";
3
4  dofile(lp.GetPath().."/config/script/loriotinit.lua");
5  lpav.Insert(10,30,99,99,LPAV_OBJ_FILL_CIRCLE);
6  lpav.SetRef(1,"ref1");
7  lpav.SetRef(2,"ref2");
8  lpav.SetRef(3,"ref3");
9  lpav.SetRef(4,"ref4");
10
11 lp.Trace(lpav.GetRef(1));
12 lp.Trace(lpav.GetRef(2));
13 lp.Trace(lpav.GetRef(3));
14 lp.Trace(lpav.GetRef(4));
15
16 lpav.RefreshMap();
17 lp_buffer = "ok";
```

Below the script editor is a table showing the current state of variables:

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

Two auxiliary windows are visible: "LoriotPro - Script Editor Trace Window" containing a list of references [ref1], [ref2], [ref3], and [ref4]; and "LoriotPro - Script Editor Activ..." which displays a large empty oval shape.

lpav.Delete

```
lpav.Delete();
```

Explication

Supprime l'objet graphique part défaut.

Attention

L'objet n'est pas supprimé de la mémoire mais marqué « deleted » et n'est plus affiché. Il sera réellement supprimé avec les autres objets graphiques marqué « deleted » en sortant du script. Si votre script crée un nombre important d'objet que vous désirez supprimer (boucle de programme) la mémoire ne sera libérée qu'à la sortie du script. L'usage de cette commande n'est pas recommandé.

Paramètres en entrée

Paramètres en sortie

Exemple

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

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,5 do
    lpav.Insert(20*i,1,9,9,zz);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
lpav.SetSelected();

if (i==3) then
lpav.Delete();
end;
end
lpav.RefreshMap();
lp_buffer = "ok";
```

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

```
3  lp_buffer = "error";
4
5  dofile(lp.GetPath().."/config/script/loriotinit.lua");
6
7  zz=LPAV_OBJ_ANALOGIC_METER_01;
8
9  -- Créé des objects r s
10 k=0;
11 -- for i=0,5 do
12     lpav.Insert(20*i,1,9,9,zz);
13 lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.
14 lpav.SetSelected());
15
16 -- if (i==3) then
17     lpav.Delete();
18 end;
19 end
20 lpav.RefreshMap();
21 lp_buffer = "ok";
```

Below the script editor, there is a table with the following data:

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65

On the right side of the interface, there are two windows: "LoriotPro - Script Editor Trace Window" and "LoriotPro - Script Editor ActiveView Si...". The ActiveView window shows a graphical representation of five analog meters, with the last one on the right highlighted with a red border and a selection cursor.

Ipav.DeleteAll

Ipav.DeleteAll();

Explication

DeleteAll supprime tous les objets graphiques de l'ActiveView.

Paramètres en entrée

Paramètres en sortie

Exemple

```
Ip_value = 0;
Ip_buffer = "error";
Ipav.Load(Ip.GetPath().."/config/active-view/sample/Cisco3500.cmp");
Ipav.RefreshMap();
Ip.Break("continue");
Ipav.DeleteAll();
Ipav.RefreshMap();
Ip_buffer = "ok";
```

Ipav.SetSelected

Ipav.SetSelected() ;

Explication

L'objet graphique par défaut est sélectionné. Cela permet de faire des sélections de groupe dans des boucles de programmation (itérations).

Paramètres en entrée

Paramètres en sortie

Exemple

```
Ip_value = 0;
Ip_buffer ="error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,5 do
    Ipav.Insert(20*i,1,9,9,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");
Ipav.SetSelected();
end
Ipav.RefreshMap();
Ip_buffer ="ok";
```

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

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  dofile(lp.GetPath().."/config/script/loriotinit.lua");
6
7  zz=LPAV_OBJ_ANALOGIC_METER_01;
8
9  -- Créé des objects r s
10 k=0;
11 for i=0,5 do
12     lpav.Insert(20*i,1,9,9,zz);
13     lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
14     lpav.SetSelected();
15 end
16 lpav.RefreshMap();
17 lp_buffer = "ok";
```

Below the script editor, there is a table with the following data:

Value Name	Value in	
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		

To the right of the script editor, the ActiveView Simulator window shows a visual representation of the script's output, displaying six analog meter icons arranged horizontally. The first five meters are green, and the sixth is red, indicating a transition in the script's state.

Ipav.ClearSelected

Ipav.ClearSelected();

Explication

L'objet graphique par défaut est libéré « unselected ». Cela permet de faire des sélections de groupe dans des boucles de programmation (itération).

Paramètres en entrée

Paramètres en sortie

Exemple

```
Ip_value = 0;
Ip_buffer ="error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,5 do
    Ipav.Insert(20*i,1,9,9,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");
Ipav.SetSelected();
if (i==3) then
Ipav.ClearSelected();
Ipav.SetPosition( 20*i,20);
end;
end
Ipav.RefreshMap();
Ip_buffer ="ok";
```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\lpav.ClearSelected01.lua

```

1
2   lp_value = 0;
3   lp_buffer = "error";
4
5   dofile(lp.GetPath().."/config/script/loriotinit.lua");
6
7   zz=LPAV_OBJ_ANALOGIC_METER_01;
8
9   -- Créé des objects r s
10  k=0;
11  -for i=0,5 do
12    lpav.Insert(20*i,1,9,9,zz);
13    lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.
14    lpav.SetSelected();
15    -if (i==3) then
16      lpav.ClearSelected();
17      lpav.SetPosition( 20*i,20);
18    end;
19  end
20  lpav.RefreshMan();

```

LoriotPro - Script Editor Trace Win

LoriotPro - Script Editor ActiveView Simulator Window

Value Name	Value in	Value
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp...		

Ipav.ClearAllSelected

`lpav.ClearAllSelected`

Explication

Tous les objets graphique sélectionnés deviendront non sélectionnés.

Paramètres en entrée

Paramètres en sortie

Exemple

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

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,5 do
    lpav.Insert(20*i,1,9,9,zz);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
lpav.SetSelected();
if (i==3) then

lpav.SetPosition( 20*i,20);
end;
end
lpav.ClearAllSelected();
lpav.RefreshMap();
lp_buffer = "ok";
```

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

```
4  
5 dofile(lp.GetPath().."/config/script/loriotinit.lua");  
6  
7 zz=LPAV_OBJ_ANALOGIC_METER_01;  
8  
9 -- Créé des objects r s  
10 k=0;  
11 -- for i=0,5 do  
12     lpav.Insert(20*i,1,9,9,zz);  
13     lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.  
14     lpav.SetSelected();  
15 -- if (i==3) then  
16  
17     lpav.SetPosition( 20*i,20);  
18 end;  
19 end  
20 lpav.ClearAllSelected();  
21 lpav.RefreshMap();  
22 lp_buffer ="ok";
```

Below the script editor is a table with the following data:

Value Name	Value in	
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

To the right of the script editor, there are two windows: "LoriotPro - Script Editor Trace W" and "LoriotPro - Script Editor ActiveView Simulator Window". The simulator window displays a grid of six analog meter icons, with one meter icon positioned below the others.

Ipav.DeleteAllSelected

```
number=lpav.DeleteAllSelected();
```

Explication

Tous les objets graphiques sélectionnés seront effacés.

Attention

Les objets graphiques ne seront pas supprimés de la mémoire mais marqués « deleted » et plus affichés. Ils ne seront réellement supprimés avec les autres objets graphiques marqué « deleted » qu'en sortant du script. Si votre script crée un nombre important d'objets graphiques que vous désirez supprimer (boucle de programme) la mémoire ne sera libérée qu'à la sortie du script. L'usage de cette commande n'est pas recommandé.

Paramètres en entrée

Paramètres en sortie

Number Le nombre d'objets effacé.

Exemple

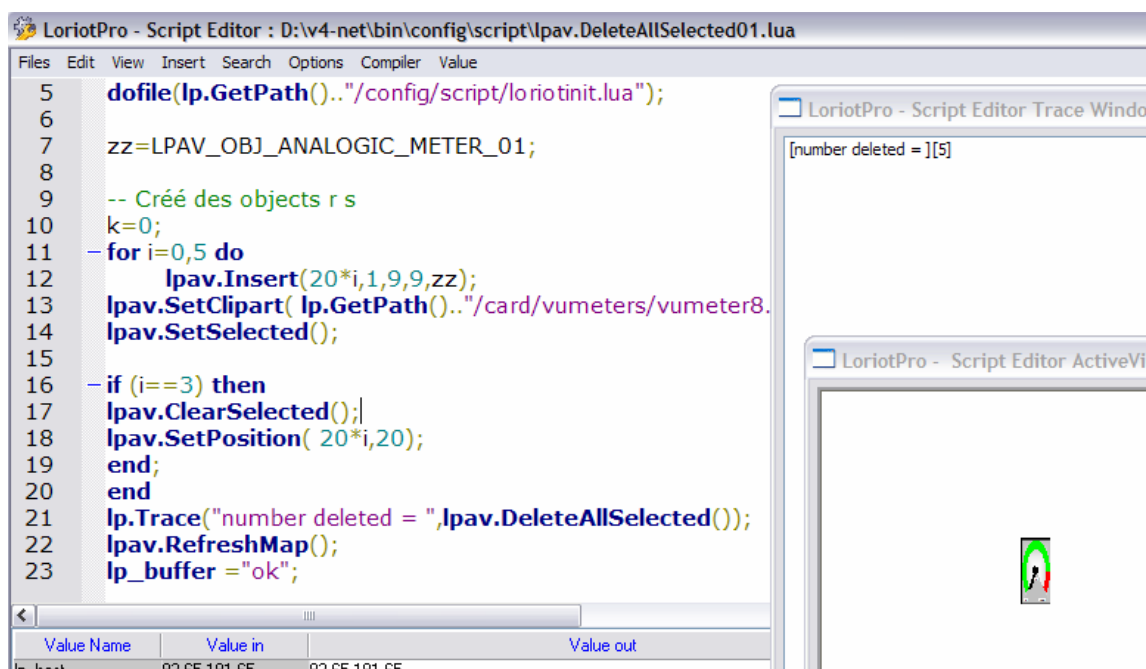
```
lp_value = 0;
lp_buffer ="error";

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

-- Créé des objects r s
k=0;
for i=0,5 do
    lpav.Insert(20*i,1,9,9,zz);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
lpav.SetSelected();

if (i==3) then
lpav.ClearSelected();
lpav.SetPosition( 20*i,20);
end;
end
lp.Trace("number deleted = ",lpav.DeleteAllSelected());
lpav.RefreshMap();
lp_buffer ="ok";
```



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

```
5 dofile(lp.GetPath().."/config/script/loriotinit.lua");
6
7 zz=LPAV_OBJ_ANALOGIC_METER_01;
8
9 -- Créé des objets r s
10 k=0;
11 --for i=0,5 do
12     lpav.Insert(20*i,1,9,9,zz);
13     lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.
14     lpav.SetSelected();
15
16 --if (i==3) then
17     lpav.ClearSelected();
18     lpav.SetPosition( 20*i,20);
19 end;
20 end
21 lp.Trace("number deleted = ",lpav.DeleteAllSelected());
22 lpav.RefreshMap();
23 lp_buffer ="ok";
```

Below the script editor is a table with the following structure:

Value Name	Value in	Value out
lp_buffer	82.65.181.65	82.65.181.65

On the right side, there are two trace windows. The top one, titled "LoriotPro - Script Editor Trace Window", shows the output: "[number deleted =][5]". The bottom one, titled "LoriotPro - Script Editor ActiveView", shows a small icon of a person.

Ipav.SelectAllRef

```
number=Ipav.SelectAllRef(number (1-4), mode, "string");
```

Explication

SelectAllRef permet de faire une sélection multiple d'objets graphiques en utilisant une variable de référence de l'objet graphique. Un objet graphique possède 4 variables de référence. Ces variables sont utilisées pour créer des groupes d'objets graphiques ou pour un usage de référencement d'objets. Si string est égal à la variable de référence alors l'ensemble des objets graphiques répondant à la référence « string » seront sélectionnés.

Paramètres en entrée

Number le numero de la référence 1 2 3 4
Mode 1 nocase sensitive 0 case sensitive
« string » une chaine de caractères de recherche

Paramètres en sortie

Number le nombre d'objet graphique sélectionné.
0 si zero ou nil.

Exemple

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
for i=0,5 do
lpav.Insert(20*i,1,9,9,zz);
lpav.SetRef(1,"ref");
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
    if (i==3) then
        lpav.SetRef(1,"other");
    end;
end
aa=lpav.SelectAllRef(1,0,"ref");
lp.Trace("number found =",aa);
if aa then
lpav.DeleteAllSelected();
end

lpav.RefreshMap();
lp_buffer ="ok";
```

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

```
7  zz=LPAV_OBJ_ANALOGIC_METER_01;
8
9  k=0;
10 -for i=0,5 do
11  lpav.Insert(20*i,1,9,9,zz);
12  lpav.SetRef(1,"ref");
13  lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
14  - if (i==3) then
15    lpav.SetRef(1,"other");
16  end;
17 end
18 aa=lpav.SelectAllRef(1,0,"ref");
19 lp.Trace("number found =",aa);
20 -if aa then
21  lpav.DeleteAllSelected();|
22 end
23
24 lpav.RefreshMap();
25 lp_buffer ="ok";
```

Below the script editor, a table shows the execution results:

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65

Two smaller windows are also visible: one titled "LoriotPro - Script Editor T..." containing the text "[number found =]5", and another titled "LoriotPro - Script Editor ActiveView Si..." which is currently empty.

Ipav.UnSelectAllRef

```
number=Ipav.UnSelectAllRef(number (1-4), mode, "string");
```

Explication

UnSelectAllRef permet de faire une désélection multiple d'objets graphiques en utilisant une variable de référence de l'objet graphique. Un objet graphique possède 4 variables de référence. Ces variables sont utilisées pour créer des groupes d'objets graphiques ou pour un usage de référencement d'objets. Si string est égal à la variable de référence alors l'ensemble des objets graphiques répondant à la référence « string » sont sélectionnés.

Paramètres en entrée

Number	le numero de la référence 1 2 3 4
Mode	1 nocase sensitive 0 case sensitive
« string »	une chaine de caractères de recherche

Paramètres en sortie

Number le nombre d'objet graphique désélectionné.
0 si zero ou nil.

Ipav.FindFromAssociatedUID

```
“uid”=Ipav.FindFromAssociatedUID( “uid-in”);
```

Explication

Cette fonction permet de trouver un objet associé à un autre (collector) par son uid.

Paramètres en entrée

« uid-in » l'uid de l'objet recherché

Paramètres en sortie

« uid » l'uid de l'objet graphique en relation avec l'objet d'uid « uid-in ».

Exemple

Ipav.SelectFirst

```
Ipav.SelectFirst();
```

Explication

Le premier objet de la liste d'objet graphique en mémoire devient l'objet graphique par défaut. Cette fonction permet de faire des boucles de programmation (itérations).

Paramètres en entrée

Paramètres en sortie

Exemple

```

Ip_value = 0;
Ip_buffer = "error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

Ipav.Load(Ip.GetPath().."/config/active-view/sample/cisco3500.cmp");
Ipav.RefreshMap();

function displayobjectInformation (co)

if Ipav.GetInformation("a") then
Ip.Trace("name>"..a.name);
Ip.Trace("Object Type ----->");
if a.objtype==LPAV_OBJ_WORLD then Ip.Trace("directory world (emf)")
elseif a.objtype==LPAV_OBJ_COUNTRY then Ip.Trace("directory country (emf)")
elseif a.objtype==LPAV_OBJ_ORGANIZATION then Ip.Trace("directory organization (emf)")
elseif a.objtype==LPAV_OBJ_ORGANIZATION_UNIT then Ip.Trace("directory organization unit (emf)")
elseif a.objtype==LPAV_OBJ_NETWORK then Ip.Trace("directory network (emf)")
elseif a.objtype==LPAV_OBJ_HOST_1 then Ip.Trace("directory host 1 (emf)")
elseif a.objtype==LPAV_OBJ_ROUTER then Ip.Trace("directory router (emf)")
elseif a.objtype==LPAV_OBJ_SHORTCUT_1 then Ip.Trace("directory shortcut 1(emf)")
elseif a.objtype==LPAV_OBJ_SHORTCUT_2 then Ip.Trace("directory shortcut 2(emf)")
elseif a.objtype==LPAV_OBJ_MRTG then Ip.Trace("directory mrtg (emf)")
elseif a.objtype==LPAV_OBJ_PLUGIN then Ip.Trace("directory plugin (emf)")
elseif a.objtype==LPAV_OBJ_BADPLUGIN then Ip.Trace("directory badplugin (emf)")
elseif a.objtype==LPAV_OBJ_LORIOTPRO then Ip.Trace("directory Loriotpro (emf)")
elseif a.objtype==LPAV_OBJ_HOST_2_ROUTER then Ip.Trace("directory host 2 router (emf)")
elseif a.objtype==LPAV_OBJ_HOST_3_SWITCH then Ip.Trace("directory host 3 switch (emf)")
elseif a.objtype==LPAV_OBJ_HOST_4 then Ip.Trace("directory host 4 (emf)")
elseif a.objtype==LPAV_OBJ_HOST_5 then Ip.Trace("directory host 5 (emf)")
elseif a.objtype==LPAV_OBJ_HOST_6 then Ip.Trace("directory host 6 (emf)")
elseif a.objtype==LPAV_OBJ_VIEWPLUGIN then Ip.Trace("directory viewplugin (emf)")
elseif a.objtype==LPAV_OBJ_HOST_SERVER then Ip.Trace("directory host server (emf)")
elseif a.objtype==LPAV_OBJ_HOST_IMPRIMANTE then Ip.Trace("directory host imprimante (emf)")
elseif a.objtype==LPAV_OBJ_HOST_PHONE then Ip.Trace("directory host phone (emf)")
elseif a.objtype==LPAV_OBJ_RECT then Ip.Trace("rectangle")
elseif a.objtype==LPAV_OBJ_CIRCLE then Ip.Trace("circle")
elseif a.objtype==LPAV_OBJ_FILL_RECT then Ip.Trace("fillrectangle")
elseif a.objtype==LPAV_OBJ_FILL_CIRCLE then Ip.Trace("fill circle")

```

```

elseif a.objtype==LPAV_OBJ_H_LINE then lp.Trace("h line")
elseif a.objtype==LPAV_OBJ_V_LINE then lp.Trace("v line")
elseif a.objtype==LPAV_OBJ_DIR_EMF then lp.Trace("directory object (emf)")
elseif a.objtype==LPAV_OBJ_TEXT then lp.Trace("texte")
elseif a.objtype==LPAV_OBJ_FILL_TEXT then lp.Trace("fill texte")
elseif a.objtype==LPAV_OBJ_FULL_RECT then lp.Trace("full rectangle")
elseif a.objtype==LPAV_OBJ_FULL_CIRCLE then lp.Trace("full circle")
elseif a.objtype==LPAV_OBJ_H45_LINE then lp.Trace("H line 45")
elseif a.objtype==LPAV_OBJ_V45_LINE then lp.Trace("V line 45")
elseif a.objtype==LPAV_OBJ_FULL_TEXT then lp.Trace("full texte")
elseif a.objtype==LPAV_OBJ_BASIC_SHAPE then lp.Trace("basic shape (emf)")
elseif a.objtype==LPAV_OBJ_ANALOGIC_METER_01 then lp.Trace("analogic meter (emf)")
elseif a.objtype==LPAV_OBJ_HOST_HUB then lp.Trace("host hub")
elseif a.objtype==LPAV_OBJ_CONNECTOR then lp.Trace("connector")
end
lp.Trace("fontname>".a.fontname);
lp.Trace("x>".a.x);
lp.Trace("y>".a.y);
lp.Trace("x1>".a.x1);
lp.Trace("y1>".a.y1);
lp.Trace("polling_interval>".a.polling_interval);
lp.Trace("lastpolling>".a.lastpolling);
lp.Trace("lastpollingresp>".a.lastpollingresp);
lp.Trace("polling>".a.polling);
lp.Trace("result>".a.result);
lp.Trace("clipartname>".a.clipartname);
lp.Trace("linewidth>".a.linewidth);
lp.Trace("objtype>".a.objtype);
lp.Trace("maptype>".a.maptype);
lp.Trace("ref1>".a.ref1);
lp.Trace("ref2>".a.ref2);
lp.Trace("ref3>".a.ref3);
lp.Trace("ref4>".a.ref4);
lp.Trace("UID>".a.UID);
return 1;
end
return 0;
end

if lpav.SelectFirst() then
co=lpav.SelectFirst();
while co~=nil do
lp.Trace("<<<object>>>");
displayobjectInformation (co);
co=lpav.SelectNext();
end
end

lp.Trace("fin");

```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\new\lpav.GetInformation.lua

```

71  lp.Trace("ref3>"..a.ref3);
72  lp.Trace("ref4>"..a.ref4);
73  lp.Trace("UID>"..a.UID);
74  return 1;
75  end
76  return 0;
77  end
78
79
80  -if lpav.SelectFirst() then
81  co=lpav.SelectFirst();
82  -while co~=nil do
83  lp.Trace("<<<object>>>");
84  displayobjectInformation (co);
85  co=lpav.SelectNext();
86  end
87  end
88
89  lp.Trace("fin");

```

LoriotPro - Script Editor Trace Window

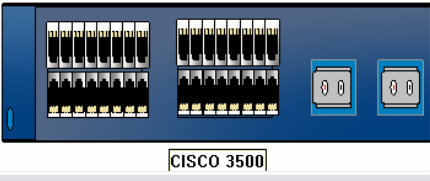
```

[<<<object>>>]
[name>CISCO 3500]
[Object Type ----->]
[directory object (emf)]
[fontname>System]
[x>2]
[y>13]
[x1>652]
[y1>66]
[polling_interval>15]
[lastpolling>0]
[lastpollingresp>0]
[polling>0]
[result>(null)]
[dispartname>card/devices/cisco/ciscocatalyst3500.emf]
[linewidth>1]
[objtype>36]
[maptype>2]
[ref1>(null)]
[ref2>(null)]
[ref3>(null)]
[ref4>(null)]
[UID>4604358336068976842]
[<<<object>>>]
[name>Connectors]
[Object Type ----->]
[directory object (emf)]
[fontname>System]
[x>67]
[y>41]
[x1>29]

```

Value Name	Value in	Value out
lp_host	82.65.181.65	
lp_index		
lp_trace	1	
lp_oid		
lp_buffer		
lp_value		

LoriotPro - Script Editor ActiveView Simulator W...



CISCO 3500

Ipav.SelectNext

```
Ipav.SelectNext();
```

Explication

Avec Ipav.SelectFirst() Le premier objet de la liste d'objet graphique en mémoire devient l'objet graphique par défaut avec Ipav.SelectNext() on sélectionne le suivant dans la liste qui devient l'objet graphique par défaut. Cette fonction permet de faire des boucles de programmation (itérations).

Paramètres en entrée

Paramètres en sortie

Exemple

Voir Ipav.SelectFirst() ;

Ipav.GetGlobalOption

Result=Lpav.GetGlobalOption("array");

Explication

Cette fonction permet de récupérer dans un tableau les paramètres globaux de l'ActiveView.

Paramètres en entrée

« array » le nom d'un tableau pour collecter les valeurs.

Paramètres en sortie

Result = 1 si le tableau est rempli ou nil si une erreur.

Variables en retour	description
arrayname ['defip']	L'adresse ip par défaut de l'ActiveView.
arrayname ['mapname']	Le nom de l'ActiveView
arrayname ['fontname']	Le nom de la fonte part défaut.
arrayname ['fontsize']	
arrayname ['fontitalic']	
arrayname ['linewidth']	
arrayname ['linestyle']	
arrayname ['maxx']	La taille max de l'ActiveView en largeur (pixel)
arrayname ['maxy']	La taille max de l'ActiveView en hauteur (pixel)
arrayname ['ZoomAuto']	1 mode zoom auto 0 utilise l'echelle
arrayname ['ZoomIndex']	
arrayname['ScaleX']	
arrayname['ScaleY']	

Exemple

```

Ip_value = 0;
Ip_buffer = "error";
-- require(lp.GetPath().."/config/script/loriotinit.lua");
if Ipav.GetGlobalOption("a") then -- place les réponses dans une array a
    Ip.Trace("defip>"..a.defip);
    Ip.Trace("mapname>"..a.mapname);
    Ip.Trace("fontname>"..a.fontname);
    Ip.Trace("fontsize>"..a.fontsize);
    Ip.Trace("fontitalic>"..a.fontitalic);
    Ip.Trace("linewidth>"..a.linewidth);
    Ip.Trace("linestyle>"..a.linestyle);
    Ip.Trace("maxx>"..a.maxx);
    Ip.Trace("maxy>"..a.maxy);
    Ip.Trace("ZoomAuto>"..a.ZoomAuto);
    Ip.Trace("ZoomIndex>"..a.ZoomIndex);

```

end

LoriotPro - Script Editor : D:\v4-net\bin\config\script\new\lpav.GetGlobalOption.lua

Files Edit View Insert Search Options Compiler Value

```

1
2   lp_value = 0;
3   lp_buffer = "error";
4   if lpav.GetGlobalOption("a") then y a
5       lp.Trace("defip>"..a.defip);
6       lp.Trace("mapname>"..a.mapname);
7       lp.Trace("fontname>"..a.fontname);
8       lp.Trace("fontsize>"..a.fontsize);
9       lp.Trace("fontitalic>"..a.fontitalic);
10      lp.Trace("linewidth>"..a.linewidth);
11      lp.Trace("linestyle>"..a.linestyle);
12      lp.Trace("maxx>"..a.maxx);
13      lp.Trace("maxy>"..a.maxy);
14      lp.Trace("ZoomAuto>"..a.ZoomAuto);
15      lp.Trace("ZoomIndex>"..a.ZoomIndex);
16
17   end
18

```

LoriotPro - Script Editor Trace Window

```

[defip>82.65.181.65]
[mapname>Cisco 3500]
[fontname>System]
[fontsize>-8]
[fontitalic>0]
[linewidth>1]
[linestyle>512]
[maxx>700]
[maxy>480]
[ZoomAuto>0]
[ZoomIndex>0]

```

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		

Ipav.SetGlobalOption

```
result=Ipav.SetGlobalOption(optionref,"value");
```

Explication

Cette fonction permet de modifier les paramètres par défaut d'une ActiveView.

Paramètres en entrée

Optionref l'option à configurer, la liste des options est incluse dans le fichier loriotinit.lua.

```
--Ipav.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 ;
```

« value » la valeur donnée à l'option

Optionref	Value
LPAV_G_DEFAULTIP	« ip » L'adresse ip par défaut pour l'ActiveView
LPAV_G_MAPNAME	« name » Le nom de l'ActiveView
LPAV_G_FONTNAME	« fonte » le nom de la fonte par défaut.
LPAV_G_FONTSIZE	
LPAV_G_FONTITALIC	« 0 » ou « 1 » 0 nom italic 1 italic
LPAV_G_LINEWIDTH	« value »
LPAV_G_LINESTYLE	« value »
LPAV_G_MAX_X	« value »
LPAV_G_MAX_Y	« value »
LPAV_G_ZOOMAUTO	« 0 » ou « 1 »

	0 nom 1 oui
LPAV_G_ZOOMINDEX	
LPAV_G_SCALEX	« float » l'échelle sur la largeur
LPAV_G_SCALEY	« float » l'échelle sur la hauteur

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

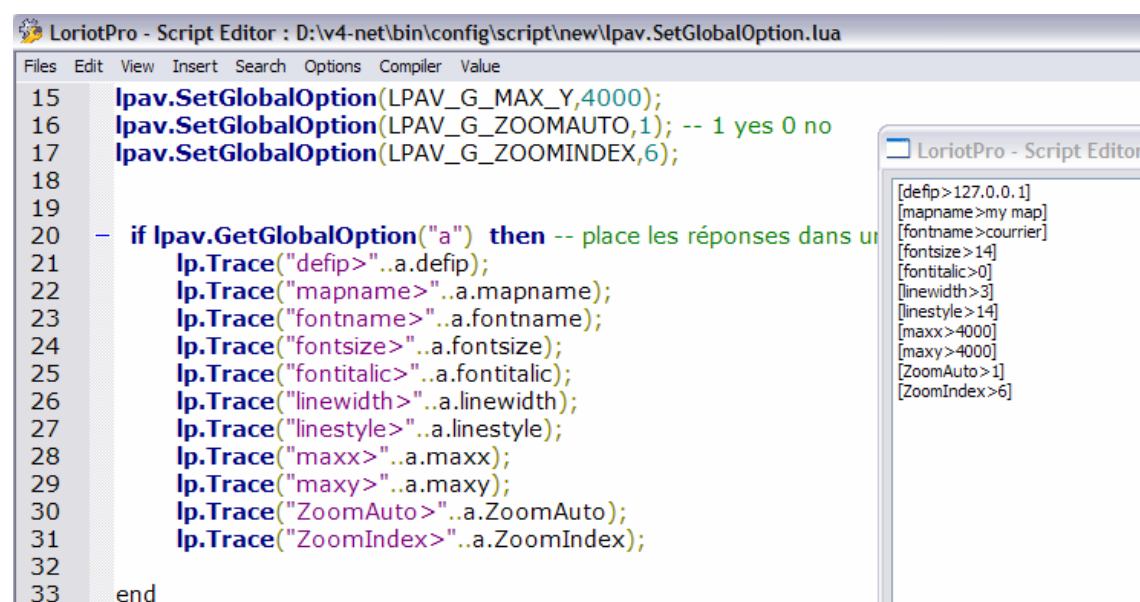
```

ip_value = 0;
ip_buffer = "error";
dofile(ip.GetPath("../config/script/loriotinit.lua");
--lpav.SetGlobalOption

lpav.SetGlobalOption(LPAV_G_DEFAULTIP,"127.0.0.1");
lpav.SetGlobalOption(LPAV_G_MAPNAME,"my map");
lpav.SetGlobalOption(LPAV_G_FONTNAME,"courrier");
lpav.SetGlobalOption(LPAV_G_FONTSIZE,"14");
lpav.SetGlobalOption(LPAV_G_FONTITALIC,"0"); -- 1 yes 0 no
lpav.SetGlobalOption(LPAV_G_LINEWIDTH,"3");
lpav.SetGlobalOption(LPAV_G_LINESTYLE,"14");
lpav.SetGlobalOption(LPAV_G_MAX_X,4000);
lpav.SetGlobalOption(LPAV_G_MAX_Y,4000);
lpav.SetGlobalOption(LPAV_G_ZOOMAUTO,1); -- 1 yes 0 no
lpav.SetGlobalOption(LPAV_G_ZOOMINDEX,6);

if lpav.GetGlobalOption("a") then -- place les réponses dans une array a
    lp.Trace("defip>".a.defip);
    lp.Trace("mapname>".a.mapname);
    lp.Trace("fontname>".a.fontname);
    lp.Trace("fontsize>".a.fontsize);
    lp.Trace("fontitalic>".a.fontitalic);
    lp.Trace("linewidth>".a.linewidth);
    lp.Trace("linestyle>".a.linestyle);
    lp.Trace("maxx>".a.maxx);
    lp.Trace("maxy>".a.maxy);
    lp.Trace("ZoomAuto>".a.ZoomAuto);
    lp.Trace("ZoomIndex>".a.ZoomIndex);
end

```

```
LoriotPro - Script Editor : D:\v4-net\bin\config\script\new\lpav.SetGlobalOption.lua
Files Edit View Insert Search Options Compiler Value
15  lpav.SetGlobalOption(LPAV_G_MAX_Y,4000);
16  lpav.SetGlobalOption(LPAV_G_ZOOMAUTO,1); -- 1 yes 0 no
17  lpav.SetGlobalOption(LPAV_G_ZOOMINDEX,6);
18
19
20  -- if lpav.GetGlobalOption("a") then -- place les réponses dans un
21      lp.Trace("defip>"..a.defip);
22      lp.Trace("mapname>"..a.mapname);
23      lp.Trace("fontname>"..a.fontname);
24      lp.Trace("fontsize>"..a.fontsize);
25      lp.Trace("fontitalic>"..a.fontitalic);
26      lp.Trace("linewidth>"..a.linewidth);
27      lp.Trace("linestyle>"..a.linestyle);
28      lp.Trace("maxx>"..a.maxx);
29      lp.Trace("maxy>"..a.maxy);
30      lp.Trace("ZoomAuto>"..a.ZoomAuto);
31      lp.Trace("ZoomIndex>"..a.ZoomIndex);
32
33  end
```

LoriotPro - Script Editor

```
[defip>127.0.0.1]
[mapname>my map]
[fontname>courrier]
[fontsize>14]
[fontitalic>0]
[linewidth>3]
[linestyle>14]
[maxx>4000]
[maxy>4000]
[ZoomAuto>1]
[ZoomIndex>6]
```

Ipav.Load

```
Result=Ipav.Load("ActiveViewfile.cmp");
```

Explication

Charge une ActiveView en effaçant la précédente.

Paramètres en entrée

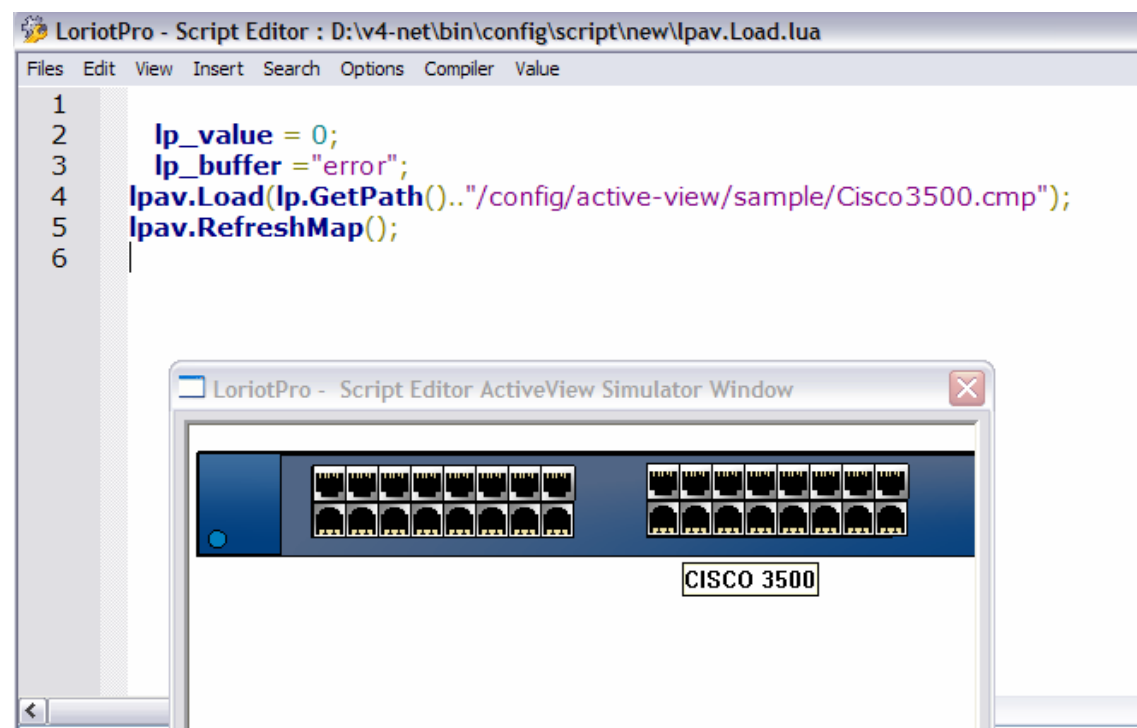
“ActiveViewfile.cmp” Le fichier contenant les paramètres à charger.

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
Ipav.Load(Ip.GetPath().."/config/active-view/sample/Cisco3500.cmp");  
Ipav.RefreshMap();
```



Ipav.Append

```
Result=Ipav.Append("ActiveViewfile.cmp");
```

Explication

Charge une extension d'ActiveView en gardant les objets graphiques déjà existant. Les objets graphiques chargés sont tous préselectionnés.

Paramètres en entrée

"ActiveViewfile.cmp" Le fichier contenant les paramètres à charger.

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer = "error";
Ipav.Append(Ip.GetPath().."/config/active-view/sample/Cisco3500.cmp");
Ipav.RefreshMap();
```

Ipav.Save

```
Result=Ipav.Save("ActiveViewfile.cmp");
```

Explication

Sauvegarde le contenu de l'ActiveView dans un fichier.

Attention si le fichier existe il est écrasé.

Paramètres en entrée

"ActiveViewfile.cmp" Le nom du fichier de sauvegarde.

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer = "error";
Ipav.Load(Ip.GetPath().."/config/active-view/sample/cisco3500.cmp");
Ipav.Save(Ip.GetPath().."/config/active-view/sample/save01.cmp");
Ip_buffer = "ok";
```

Ipav.ShiftPositionAllSelected

```
number=Ipav.ShiftPositionAllSelected(x,y) ;  
number=Ipav.ShiftPositionAllSelected(x,y,x1,y1) ;
```

Explication

Cette commande permet de déplacer de x , y l'ensemble des objets graphiques sélectionnés et éventuellement de modifier la taille des objets.

Attention :

Les objets graphiques ne peuvent pas avoir de coordonnées négatives. X Y correspond au coin supérieure gauche des objets.

Paramètres en entrée

X le delta de déplacement horizontal
Y le delta de déplacement vertical

X1 le delta de taille horizontal
Y1 le delta de taille vertical

Paramètres en sortie

Number le nombre d'objets graphiques modifié.

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
Ipav.Append(Ip.GetPath().."/config/active-view/sample/Cisco3500.cmp");  
Ipav.RefreshMap();  
Ip.Break("pause");  
Ipav.ShiftPositionAllSelected(100,100);  
Ipav.RefreshMap();  
Ip_buffer = "Ok";
```

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

```
1  
2     lp_value = 0;  
3     lp_buffer = "error";  
4     lpav.Append(lp.GetPath().."/config/active-view/sample/Cisco3500.cmp")  
5     lpav.RefreshMap();  
6     lp.Break("pause");  
7     lpav.ShiftPositionAllSelected(100,100);  
8     lpav.RefreshMap();  
9     lp_buffer = "Ok";  
10
```

Below the script editor is a table with the following data:

Value Name	Value in	
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		Ok
lp_value		0

Overlaid on the script editor is a smaller window titled "LoriotPro - Script Editor ActiveView Simulat...". This window displays a graphical simulation of a Cisco 3500 switch, showing its physical components like ports and a console. The text "CISCO 3500" is visible below the switch graphic.

Ipav.ShiftPosition

```
result=Ipav.ShiftPosition (x,y) ;  
result=Ipav.ShiftPosition (x,y,x1,y1) ;
```

Explication

Cette commande permet de déplacer de x, y, l'objet graphique sélectionné par défaut et éventuellement de modifier la taille de l'objet.

Attention :

Les objets graphiques ne peuvent pas avoir de coordonnées négatives. X Y correspond au coin supérieur gauche de l'objet.

Paramètres en entrée

X le delta de déplacement horizontal
Y le delta de déplacement vertical

X1 le delta de taille horizontal
Y1 le delta de taille vertical

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```
Ip_value = 0;  
Ip_buffer = "error";  
  
dofile(Ip.GetPath().."/config/script/loriotinit.lua");  
  
zz=LPAV_OBJ_ANALOGIC_METER_01;  
  
-- Créé des objects r s  
k=0;  
for i=0,20 do  
    Ipav.Insert(10*i,1,9,9,zz);  
    Ipav.SetBrush(0,255,0);  
    Ipav.SetFontName("Mistral");  
    Ipav.SetName(i);  
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");  
Ipav.SetOption(LPAV_UNIT,"%");  
    if i==5 then Ipav.ShiftPosition(0,20)  
    else  
        Ipav.ShiftPosition(10,10*i);  
    end  
end  
Ipav.RefreshMap();  
Ip_buffer = "ok";
```

LoriotPro - Script Editor : D:\v4-net\bin\config\script\new\lpav.ShiftPosition01.lua

```

1  lp_value = 0;
2  lp_buffer = "error";
3
4  dofile(lp.GetPath().."/config/script/loriotinit.lua");
5
6  zz=LPAV_OBJ_ANALOGIC_METER_01;
7
8  -- Créé des objects r s
9  k=0;
10 - for i=0,20 do
11     lpav.Insert(10*i,1,9,9,zz);
12     lpav.SetBrush(0,255,0);
13     lpav.SetFontName("Mistral");
14     lpav.SetName(i);
15     lpav.SetClipart(lp.GetPath().."/card/vumeters/vumeter8.emf");
16     lpav.SetOption(LPAV_UNIT,"%");
17     if i==5 then lpav.ShiftPosition(0,20)
18     else
19         lpav.ShiftPosition(10,10*i);
20     end
21 end
22 lpav.RefreshMap();
23 lp_buffer = "ok";

```

Value Name | Value in | Value out

Value Name	Value in	Value out
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

LoriotPro - Script Editor ActiveView Simulator Window

Ipav.GetRectAllSelected

```
x,y,x1,y1=Ipav.GetRectAllSelected();
```

Explication

Cette commande permet de déplacer et de récupérer le rectangle inscrit de l'ensemble des objets graphiques sélectionnés.

Attention :

Les objets graphiques ne peuvent pas avoir de coordonnées négatives. X Y correspond au coin supérieur gauche de l'objet.

Paramètres en entrée

Paramètres en sortie

X Y coin supérieur gauche du rectangle inscrit.

X1 Y1 coin inférieur droit du rectangle inscrit.

Ou nil si il n'y a pas d'objet sélectionné.

Exemple

```
Ip_value = 0;
Ip_buffer = "error";
Ipav.Append(Ip.GetPath().."/config/active-view/sample/Cisco3500.cmp");
Ipav.RefreshMap();
x,y,x1,y1=Ipav.GetRectAllSelected();
Ip.Trace("Rectangle = "..x.."/"..y.."/"..x1.."/"..y1);
Ip.Break("pause");
Ipav.ShiftPositionAllSelected(100,100,5,5);
x,y,x1,y1=Ipav.GetRectAllSelected();
Ip.Trace("Rectangle = "..x.."/"..y.."/"..x1.."/"..y1);
Ipav.RefreshMap();
Ip_buffer = "Ok";
```

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

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4  lpav.Append(lp.GetPath().."/config/active-view/sample/Cisco3500.cmp")
5  lpav.RefreshMap();
6  x,y,x1,y1=lpav.GetRectAllSelected();
7  lp.Trace("Rectangle = "..x.." / "..y.." / "..x1.." / "..y1");
8  lp.Break("pause");
9  lpav.ShiftPositionAllSelected(100,100,5,5);
10 x,y,x1,y1=lpav.GetRectAllSelected();
11 lp.Trace("Rectangle = "..x.." / "..y.." / "..x1.." / "..y1");
12 lpav.RefreshMap();
13  lp_buffer = "Ok";
14
```

Below the script editor, there is a 'Value Name' table and a simulation window:

Value Name
lp_host
lp_index
lp_trace
lp_oid
lp_buffer
lp_value

The simulation window, titled 'LoriotPro - Script Editor ActiveView Simulat...', shows a graphical representation of a Cisco 3500 switch. The switch is depicted as a rack-mounted device with multiple ports and a central control panel. The text 'CISCO 3500' is displayed below the device.

On the right side of the interface, there is a 'Trace' window titled 'LoriotPro - Script Editor Trace V'. It displays the following trace output:

```
[Rectangle =12/23/664/106]
Break at line [pause] Continue ?
[Rectangle =112/123/769/211]
```

Ipav.SetOptionAllSelected

```
number=Ipav.SetOptionAllSelected(optionref, 'value') ;
```

Explication

Cette fonction permet de modifier les paramètres de l'ensemble des objets graphiques sélectionnés.

Paramètres en entrée

Optionref le paramètre à modifier. La liste des paramètres disponibles est incluse dans le fichier loriotinit.lua

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

« value » la valeur du nouveau paramètre

Voir la commande Ipav.SetOption pour la liste des options disponibles.

Paramètres en sortie

Number le nombre d'objets graphiques modifiés
Sinon nil

Exemple

```
lp.Trace("start");
dofile(lp.GetPath().."/config/script/loriotinit.lua");
lp_value = 0;
lp_buffer ="error";
lpav.Load(lp.GetPath().."/config/active-view/sample/cisco3500.cmp");
co=lpav.SelectFirst();
while co~=nil do
    lpav.SetSelected();
co=lpav.SelectNext();
end
lpav.SetOptionAllSelected(LPAV_X1,"10");
lpav.RefreshMap();
lp.Trace("fin");
```

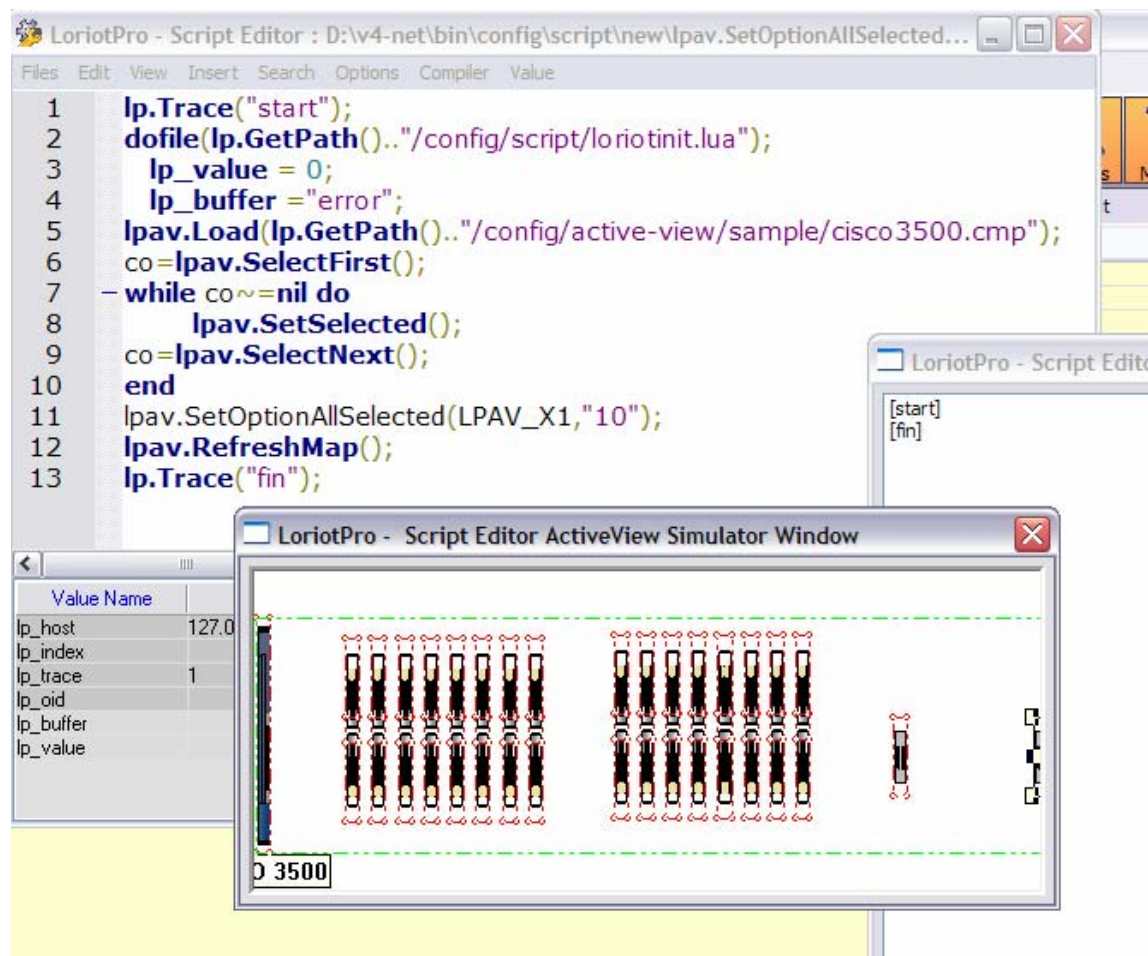


Tableau de définition de la librairie

Ipav.SelectAll

Result=Ipav.SelectAll();

Explication

Cette fonction permet de sélectionner l'ensemble des objets graphiques présents dans l'ActiveView.

Paramètres en entrée

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer ="error";
Ipav.Load(Ip.GetPath().."/config/active-view/sample/Cisco3500.cmp");
Ipav.SelectAll();
Ipav.ShiftPositionAllSelected(10,10);
Ipav.RefreshMap();
```

lpav.Copy

```
result=lpav.Copy();
```

Explication

Copie l'objet graphique part défaut.

Paramètres en sortie

Result =1 si ok sinon nil

Exemple

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

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
lpav.Insert(10,1,9,9,zz);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
lpav.Insert(20,1,9,9,zz);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter12.emf");
lpav.Copy();

lpav.DeleteAll();
lpav.Paste(0);
lp_buffer ="ok";
lp.Trace("fin");
lpav.RefreshMap();
```

Ipav.CopyAllSelected

```
result=Ipav.CopyAllSelected();
```

Explication

Copie l'ensemble des objets graphiques sélectionnés.

Paramètres en sortie

Result =1 si ok sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer ="error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
Ipav.Insert(10,1,9,9,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");
Ipav.SetSelected();
Ipav.Insert(20,1,9,9,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter12.emf");
Ipav.SetSelected();
Ipav.CopyAllSelected();
Ipav.ClearAllSelected();
Ipav.DeleteAll();
Ipav.Paste(0);

--for i=0,5 do
--
--Ipav.ShiftPositionAllSelected(10*i,10);
--end

Ip.Trace("fin");
Ipav.RefreshMap();
```

Ipav.Paste

```
result=Ipav.Paste(mode) ;
```

Explication

Colle l'objet graphique ou l'ensemble des objets graphiques copiés.

Paramètres en entrée

Mode 0 en mode append (tous les objets graphiques chargés sont sélectionnés)
1 en mode load (les objets graphiques chargés ne sont pas sélectionnés) ;

Paramètres en sortie

Result =1 si ok sinon nil

Exemple

```
Ip_value = 0;
Ip_buffer = "error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
Ipav.Insert(10,1,9,9,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");
Ipav.Insert(20,1,9,9,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter12.emf");
Ipav.Copy();

Ipav.DeleteAll();
Ipav.Paste(0);
Ip_buffer = "ok";
Ip.Trace("fin");
Ipav.RefreshMap();
```


Ipav.SetupCornerAllSelected

number=Ipav.SetupCornerAllSelected(x,y) ;

Explication

Place le coin supérieur gauche du rectangle inscrit des objets graphiques sélectionnés en x, y.

Paramètres en entrée

X,Y Le nouveau coin supérieur gauche du rectangle inscrit des objets graphiques sélectionnés.

Paramètres en sortie

Number le nombre d'objets graphiques déplacés.
Ou nil si une erreur

Exemple

```
Ip_value = 0;
Ip_buffer = "error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
Ipav.Insert(10,1,10,10,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");
Ipav.SetSelected();
Ipav.Insert(20,1,20,20,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter12.emf");
Ipav.SetSelected();
Ipav.CopyAllSelected();
Ipav.DeleteAll();

Ipav.Paste(0);
Ipav.SetupCornerAllSelected(100,10);
Ipav.RefreshMap();
Ip.Break("pause");
Ipav.ClearAllSelected();
Ipav.Paste(0);
Ipav.SetupCornerAllSelected(200,10);
Ipav.RefreshMap();
Ip.Break("pause");
Ipav.ClearAllSelected();
Ipav.Paste(0);
Ipav.SetupCornerAllSelected(300,10);
Ipav.RefreshMap();
Ip.Break("pause");

Ip_buffer = "ok";
```

```
lp.Trace("fin");
lpav.RefreshMap();
```

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

```

10 lpav.Insert(10,1,10,10,zz);
11 lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
12 lpav.SetSelected();
13 lpav.Insert(20,1,20,20,zz);
14 lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");
15 lpav.SetSelected();
16 lpav.CopyAllSelected();
17 lpav.DeleteAll();
18
19 lpav.Paste(0);
20 lpav.SetUpCornerAllSelected(100,10);
21 lpav.RefreshMap();
22 lp.Break("pause");
23 lpav.ClearAllSelected();
24 lpav.Paste(0);
25 lpav.SetUpCornerAllSelected(200,10);
26 lpav.RefreshMap();
27 lp.Break("pause");
28 lpav.ClearAllSelected();
29 lpav.Paste(0);
30 lpav.SetUpCornerAllSelected(300,10);
31 lpav.RefreshMap();
32 lp.Break("pause");
33
34 lp_buffer ="ok";
35 lp.Trace("fin");
36 lpav.RefreshMap();
37

```

Below the script is a table with the following data:

Value Name	Value in	Value
lp_host	82.65.181.65	82.65.181.65
lp_index		
lp_inace	1	1
lp_oid		
lp_buffer	ok	
lp_value	0	

Overlaid on the script editor are two other windows:

- LoriotPro - Script Editor Trace Window:** This window shows a list of break points with checkboxes for 'Continue?'. The first three entries are:
 - Break at line [pause] Continue ?
 - Break at line [pause] Continue ?
 - Break at line [pause] Continue ?
- LoriotPro - Script Editor ActiveView Simulator Window:** This window displays a graphical representation of three meters or gauges, each with a needle and a scale, arranged horizontally.

Ipav.SizeInRectAllSelected

number=Ipav.SizeInRectAllSelected(x,y,x1,y1);

Explication

Cette fonction permet de positionner avec mise à l'échelle l'ensemble des objets graphiques sélectionnés dans un rectangle inscrit de l'ActiveView.

Attention les coordonnées du rectangle doivent être bonnes et la largeur ($x1 > (x+5)$) ou la hauteur ($y1 > (y+5)$) supérieures à 5. Du fait de la conversion de double en entier dans le calcul d'échelle, cette fonction n'est pas très précise mais elle permet de positionner très simplement des ensembles d'objets graphiques très rapidement.

Paramètres en entrée

X,Y Les coordonnées du coin supérieur gauche du rectangle
X1,Y1 Les coordonnées du coin inférieure droit du rectangle.

Paramètres en sortie

Number Le nombre d'objets Graphiques déplacés.
nil si une erreur

Exemple

```
Ip_value = 0;
Ip_buffer ="error";

dofile(Ip.GetPath().."/config/script/loriotinit.lua");

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
Ipav.Insert(10,1,10,10,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter8.emf");
Ipav.SetSelected();
Ipav.Insert(20,1,20,20,zz);
Ipav.SetClipart( Ip.GetPath().."/card/vumeters/vumeter12.emf");
Ipav.SetSelected();
Ipav.CopyAllSelected();
Ipav.RefreshMap();
Ip.Break("pause");

Ipav.DeleteAll();

Ipav.ClearAllSelected();
Ipav.Paste(0);
Ipav.SizeInRectAllSelected(3,3+0, 3+20,3+20+0);
Ipav.RefreshMap();
Ip.Break("pause");

Ipav.ClearAllSelected();
Ipav.Paste(0);
```

```

Ipav.SizeInRectAllSelected(3,3+20, 3+30,3+20+20);
Ipav.RefreshMap();
Ip.Break("pause");

Ipav.ClearAllSelected();
Ipav.Paste(0);
Ipav.SizeInRectAllSelected(3,3+40, 3+40,3+20+40);
Ipav.RefreshMap();
Ip.Break("pause");

Ipav.ClearAllSelected();
Ipav.Paste(0);
Ipav.SizeInRectAllSelected(3,3+60, 3+50,3+20+60);
Ipav.RefreshMap();
Ip.Break("pause");

Ipav.ClearAllSelected();
Ipav.Paste(0);
Ipav.SizeInRectAllSelected(3,3+80, 3+60,3+20+80);
Ipav.RefreshMap();
Ip.Break("pause");

Ip_buffer ="ok";
Ip.Trace("fin");
Ipav.RefreshMap();
    
```

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

```

29  Ipav.Paste(0);
30  Ipav.SizeInRectAllSelected(3,3+20, 3+30,3+20+20);
31  Ipav.RefreshMap();
32  Ip.Break("pause");
33
34  Ipav.ClearAllSelected();
35  Ipav.Paste(0);
36  Ipav.SizeInRectAllSelected(3,3+40, 3+40,3+20+40);
37  Ipav.RefreshMap();
38  Ip.Break("pause");
39
40  Ipav.ClearAllSelected();
41  Ipav.Paste(0);
42  Ipav.SizeInRectAllSelected(3,3+60, 3+50,3+20+60);
43  Ipav.RefreshMap();
44  Ip.Break("pause");
45
46  Ipav.ClearAllSelected();
47  Ipav.Paste(0);
48  Ipav.SizeInRectAllSelected(3,3+80, 3+60,3+20+80);
49  Ipav.RefreshMap();
50  Ip.Break("pause");
51
52  Ip_buffer ="ok";
53  Ip.Trace("fin");
54  Ipav.RefreshMap();
55
    
```

Below the script, a table shows the current values of variables:

Value Name	Value in	Value out
Ip_host	82.65.181.65	82.65.181.65
Ip_index		
Ip_trace	1	1
Ip_oid		
Ip_buffer		ok
Ip_value		0

On the right side, the 'ActiveView Simulation' window shows a graphical representation of a network map with several gauges and indicators. The 'Trace Window' at the bottom right shows the execution flow with break points and continuation options.

Ipav.InsertAction

```
result=Ipav.InsertAction(type,'parameters');
```

Explication

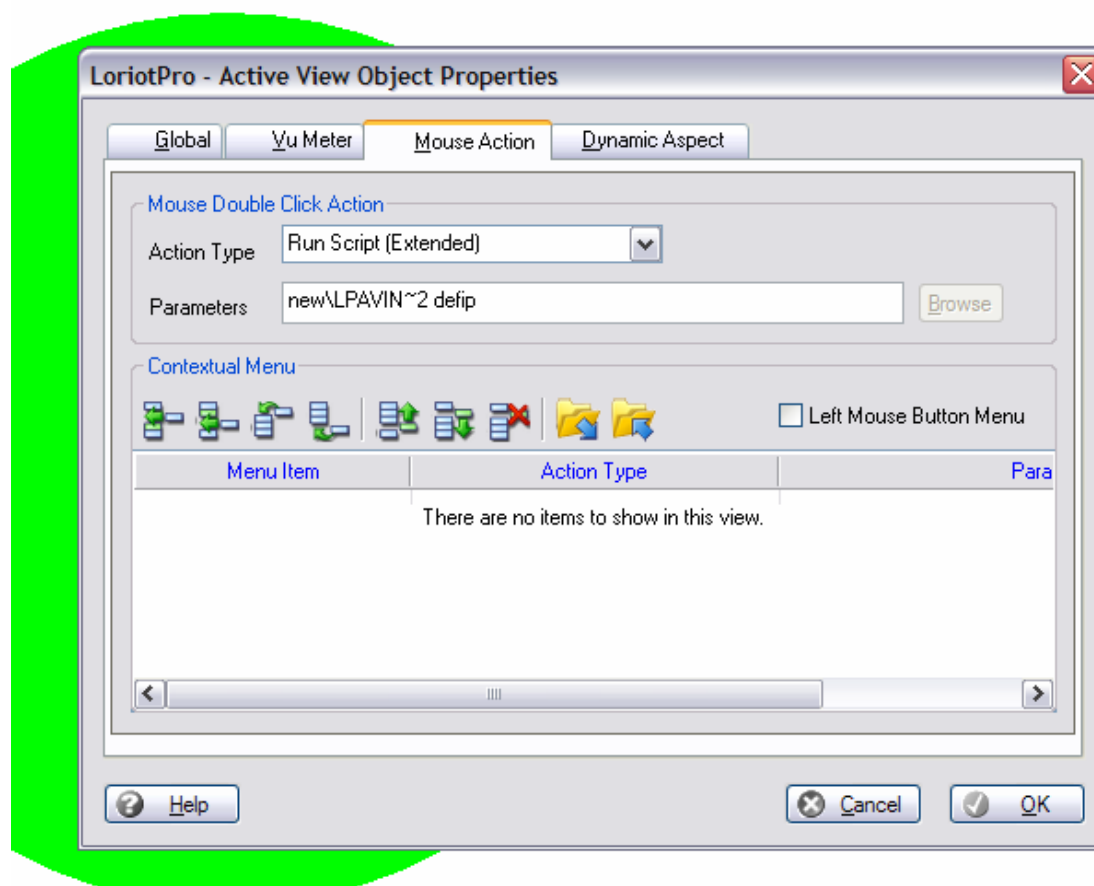
Cette fonction permet de définir l'action (double click de souris) associée à l'objet graphique par défaut.

Paramètres en entrée

Type le type d'action à réaliser. La liste des paramètres disponibles est incluse dans le fichier loriotinit.lua

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

'parameter' les paramètres liés à l'action type, il s'agit de la chaîne de caractères générée avec les modules Wizard des propriétés d'action d'un objet graphique. Utiliser ce module pour générer vos chaînes de paramètres et reportez vous a la documentation du logiciel.



Attention le logiciel ne teste pas la coherence de vos paramètres.

type	parameter
LPAV_ACTION_NONE	
LPAV_ACTION_LOADMAP	
LPAV_ACTION_WINEXEC	
LPAV_ACTION_SHELLEXEC	
LPAV_ACTION_PLUGIN	
LPAV_ACTION_TABLE	
LPAV_ACTION_SCRIPT_REP	
LPAV_ACTION_SCI	
LPAV_ACTION_FOUND_INDIR	
LPAV_ACTION_MODAL_MAP	
LPAV_ACTION_SET_OID	
LPAV_ACTION_CLEAR_EVENTREF	
LPAV_ACTION_CLEAR_EVENTFILTER	
LPAV_ACTION_CLEAR_TRAPFILTER	
LPAV_ACTION_CLEAR_EVENTGENERATED	
LPAV_ACTION_CLEAR_TRAPGENERATED	
LPAV_ACTION_CLEAR_TRAPREF	
LPAV_ACTION_RUN_SCRIPT_LUA	

Paramètres en sortie

Result = 1 si ok sinon nil

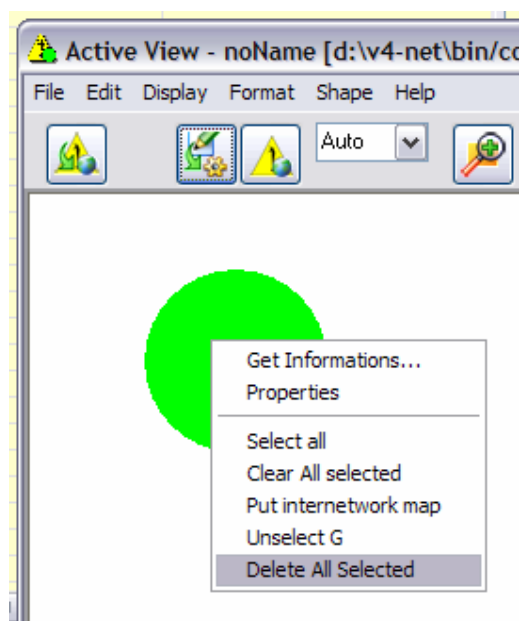
Exemple

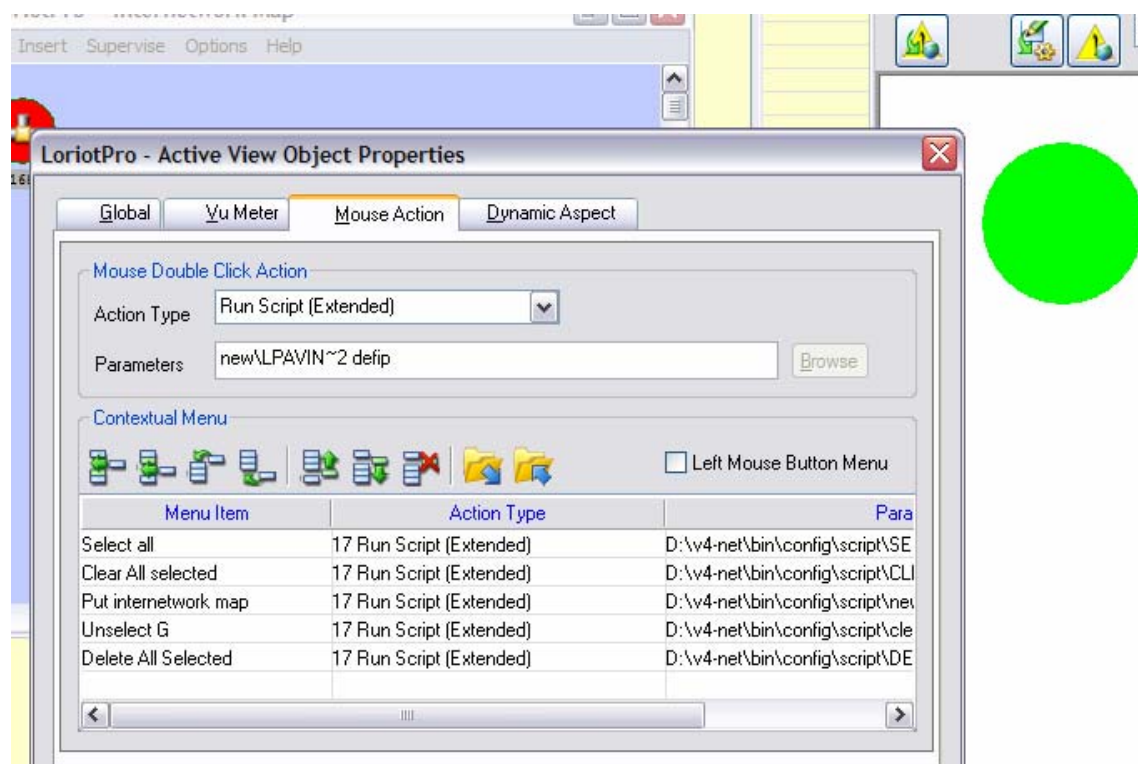
Ipav.InsertMenu

```
result=Ipav.InsertMenu(type, "caption", "parameters");
```

Explication

Cette fonction permet de créer un menu contextuel pour l'objet graphique par défaut, elle réutilise les mêmes paramètres types que la fonction InsertAction. Les nouvelles actions associées au menu contextuel de l'objet graphique sont insérées à la fin du menu.





Ces informations sont stockées dans le fichier cmp de l'activeview sous le nom

```
object menu 17 "Select all" "D:\v4-net\bin\config\script\SELECT~1.LUA defip"
object menu 17 "Clear All selected" "D:\v4-net\bin\config\script\CLEARA~1.LUA defip"
object menu 17 "Put internetwork map" "D:\v4-net\bin\config\script\new\LPAVIN~2.LUA defip"
object menu 17 "Unselect G" "D:\v4-net\bin\config\script\clean.lua defip"
object menu 17 "Delete All Selected" "D:\v4-net\bin\config\script\DELETE~1.LUA defip"
```

On peut facilement les convertir :

```
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Select all","D:\v4-net\bin\config\script\SELECT~1.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Clear All selected","D:\v4-net\bin\config\script\CLEARA~1.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Put internetwork map","D:\v4-net\bin\config\script\new\LPAVIN~2.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Unselect G","D:\v4-net\bin\config\script\clean.lua defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Delete All Selected","D:\v4-net\bin\config\script\DELETE~1.LUA defip");
```

```

global m_def_f_cStrikeOut 0
global m_def_f_nCharSet 0
global m_def_f_nOutPrecision 0
global m_def_f_nClipPrecision 0
global m_def_f_nQuality 0
global m_def_f_nPitchAndFamily 0
global m_def_m_sFontName "courier new"
#-----
object new 4861048800271141506
object x 61.000
object y 40.000
object x1 97.000
object y1 97.000
object color 13158600
object type 0
object ActionType 17
object ActionParameters "new\LPAVIN~2 defip"
object name "Filled Ellipse"
object m_iTextStyle 37
object m_sRef1 "G"
object maptype 40
object status 10
object linecolor 0
object linestyle 512
object linewidth 1
object brushcolor 65280
object objtype 40
object m_iNoIPSrcTest 0
object m_sFontName "courier new"
object f_nHeight 12
object f_nWidth 8
object f_nWeight 400
object polling_interval 15
object uid 4861048800271141506
object menu 17 "Select all" "D:\v4-net\bin\config\script\SELECT~1.LUA defip"
object menu 17 "Clear All selected" "D:\v4-net\bin\config\script\CLEARA~1.LUA defip"
object menu 17 "Put internetwork map" "D:\v4-net\bin\config\script\new\LPAVIN~2.LUA defip"
object menu 17 "Unselect G" "D:\v4-net\bin\config\script\clean.lua defip"
object menu 17 "Delete All Selected" "D:\v4-net\bin\config\script\DELETE~1.LUA defip"
    
```

Paramètres en entrée

Type le type d'action à réaliser. La liste des paramètres disponibles est incluse dans le fichier loriotinit.lua

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

'Caption ' le titre de l'action dans le menu contextuel

'parameter' les paramètres liés à l'action type, il s'agit de la chaîne de caractères générée avec les modules Wizard des propriétés d'action d'un objet graphique. Utiliser ce module pour générer vos chaînes de paramètres et reportez vous à la documentation du logiciel.

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```

lp_value = 0;
lp_buffer = "error";

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
for i=0,20 do
    lpav.Insert(10*i,1,9,9,zz);
    if (k==0) then lpav.SetRef(1,"r"); k=1; else lpav.SetRef(1,"s"); k=0; end
    lpav.SetBrush(0,255,0);
    lpav.SetFontName("Mistral");
    lpav.SetName(i);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");

lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Select all","D:\v4-
net\bin\config\script\SELECT~1.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Clear All selected","D:\v4-
net\bin\config\script\CLEARA~1.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Put internetwork map","D:\v4-
net\bin\config\script\new\LPAVIN~2.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Unselect G","D:\v4-
net\bin\config\script\clean.lua defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Delete All Selected","D:\v4-
net\bin\config\script\DELETE~1.LUA defip");

lpav.SetOption(LPAV_UNIT,"%");
end
lp.Trace("fin");
lpav.RefreshMap();

```

Ipav.DeleteAllMenu

```
result=Ipav.DeleteAllMenu();
```

Explication

Cette fonction supprime l'ensemble des menus contextuels d'un objet graphique.

Paramètres en entrée

Paramètres en sortie

Result=1 si ok sinon nil

Ipav.SetExpression

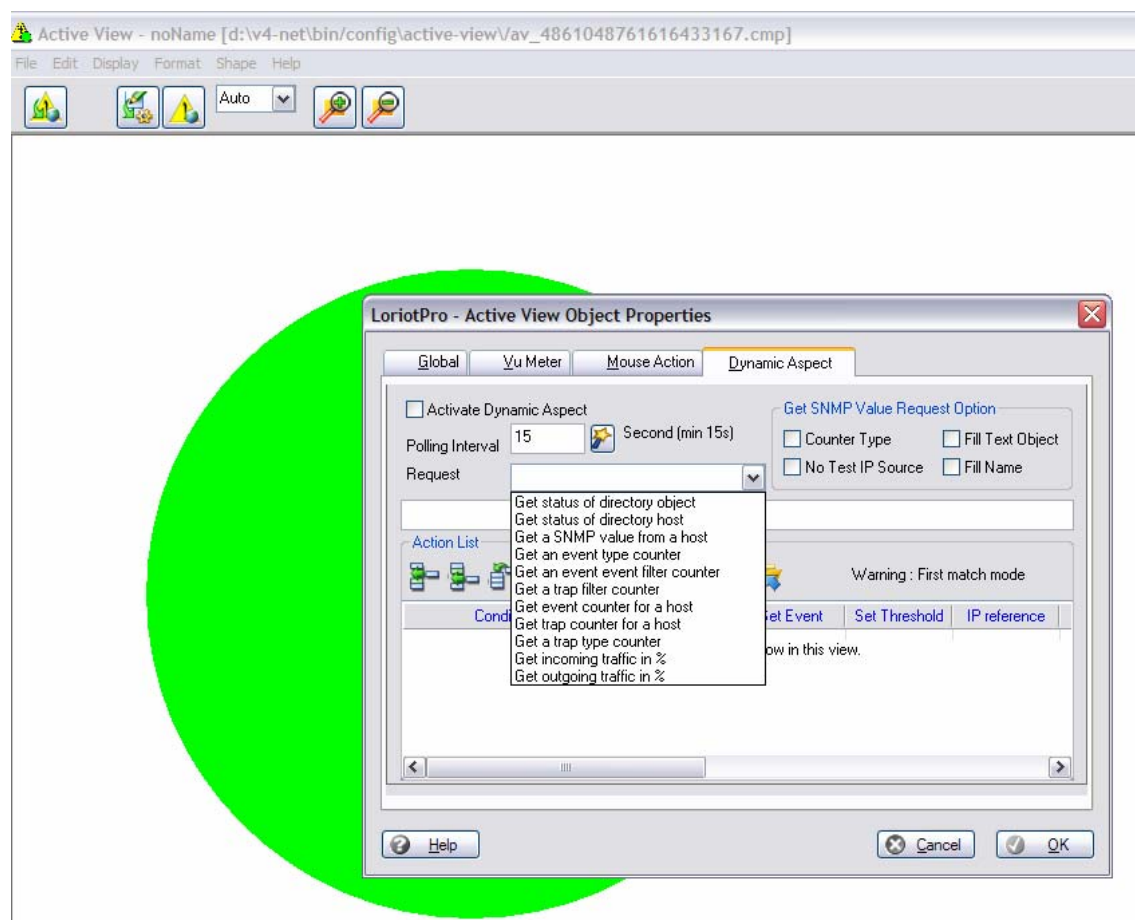
```
Result=Ipav.SetExpression("expression");
```

Explication

Cette fonction permet de définir l'expression associée à l'objet graphique par défaut.

Paramètres en entrée

« expression » Une chaîne de caractères définissant l'expression (travail de polling) à réaliser pour cet objet graphique par le scheduler de l'ActiveView. Il est possible d'utiliser les modules Wizard associés aux propriétés de l'objet pour construire la chaîne. Reportez vous a la documentation des ActiveView pour plus d'informations.



Attention le programme ne teste pas la cohérence de l'expression. Il faut impérativement configurer les paramètres de polling pour que l'expression soit déclenchée (schedulée).

```
lpav.SetOption(LPAV_POLLING_INTERVAL, « 15 »);  
lpav.SetOption(LPAV_POLLING, « 1 »);  
il est possible d'utiliser la fonction SetOption pour configurer ce parametre  
expression.
```

```
lpav.SetOption(LPAV_EXPRESSION, « getipoid defip sysname »);
```

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

```
lp_value = 0;  
lp_buffer = "error";  
  
dofile(lp.GetPath().."/config/script/loriotinit.lua");  
  
zz=LPAV_OBJ_ANALOGIC_METER_01;  
  
k=0;  
for i=0,20 do  
lpav.Insert(10*i,1,9,9,zz);  
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");  
lpav.SetOption(LPAV_UNIT,"%");  
lpav.SetOption(LPAV_POLLING_INTERVAL, "15");  
lpav.SetOption(LPAV_POLLING, "1");  
lpav.SetOption(LPAV_EXPRESSION, "getipoid defip sysname");  
lpav.GetInformation("a");  
lp.Trace(a.expression, a.polling,a.polling_interval);  
lpav.SetExpression("getipoid defip sysname");  
lpav.GetInformation("a");  
lp.Trace("set",a.expression, a.polling,a.polling_interval);  
  
end  
lp.Trace("fin");  
lpav.RefreshMap();
```

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

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  dofile(lp.GetPath().."/config/script/loriotinit.lua");
6
7  zz=LPAV_OBJ_ANALOGIC_METER_01;
8
9  -- Créé des objets r s
10 k=0;
11 for i=0,20 do
12   lpav.Insert(10*i,1,9,9,zz);
13   lpav.SetClipart(lp.GetPath().."/card/vumeters/vumeter8.emf");
14   lpav.SetOption(LPAV_UNIT,"%");
15   lpav.SetOption(LPAV_POLLING_INTERVAL,"15");
16   lpav.SetOption(LPAV_POLLING,"1");
17   lpav.SetOption(LPAV_EXPRESSION,"getipoid defip sysname");
18   lpav.GetInformation("a");
19   lp.Trace(a.expression, a.polling.a.polling_interval);
20   lpav.SetExpression("getipoid defip sysname");
21   lpav.GetInformation("a");
22   lp.Trace("set" a.expression, a.polling.a.polling_interval);
23
24 end
25 lp.Trace("fin");
26 lpav.RefreshMap();
27
```

Two auxiliary windows are visible:

- LoriotPro - Script Editor Trace Window:** Displays a list of log entries, each consisting of a 'set' or 'get' command followed by the object ID and the expression 'defip sysname'. The entries are repeated 21 times.
- LoriotPro - Script Editor ActiveView Simulat...:** Shows a graphical representation of the simulated environment with a row of 21 green vertical bars, each topped with a small icon, representing the state of the 21 objects created in the script.

At the bottom of the editor, there is a table for monitoring values:

Value Name	Value in	Value out

Ipav.DeleteAllFilter

```
result=Ipav.DeleteAllFilter() ;
```

Explication

Cette fonction supprime l'ensemble des filtres associés au résultat fourni par une requête (expression) de polling liée à un objet graphique.

Paramètres en sortie

Result=1 si ok sinon nil

Ipav.InsertFilter

Result=Ipav.InsertFilter(condition,'value',color, event,threshold,'ip','mask','msg',level);

Explication

Cette fonction permet de créer un nouveau filtre associé au résultat fourni par une requête (expression) de polling liée à un objet graphique.

Paramètres en entrée

Condition

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

```
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;
```

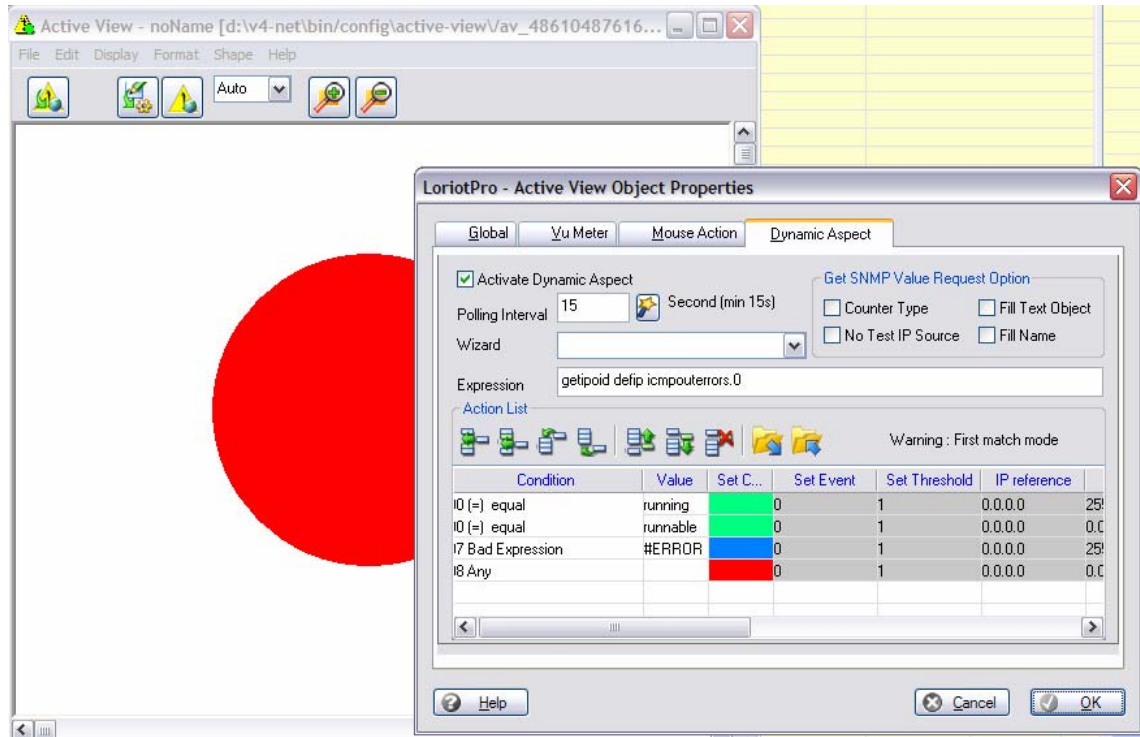
Color une valeur entière sur 32 bits au format RGB (0=noir)

'value' une chaîne de caractères correspondant au test du résultat d'un filtre
ou «(null)».

event Si supérieur a zero le numéro d'événement à envoyer à LoriotPro si la condition est validée.

Threshold tous les combien
'ip' L'adresse IP (xx.xx.xx.xx) de référence pour l'événement
'mask' le mask sous sa forme ip (xx.xx.xx.xx) pour l'événement
'msg' le message de l'événement ou «(null)»
level le level de l'événement

La fonction insert filter est complexe, vous pouvez vous aider en utilisant les modules wizard du programme de propriété d'un objet graphique.



Vous pouvez vous aider en consultant le fichier cmp de votre ActiveView.

```
object filter 0 "runnable" 8453888 0 1 0 0 "(null)" 0
object filter 7 "#ERROR" 16744448 0 1 0 4294967295 "Bad expression for [defip]" 3
object filter 8 "(null)" 255 0 1 0 0 "(null)" 0
```

Les adresses ip sont sauvegardés en mode entier 32bits, il faut donc faire la conversion

```
lpav.InsertFilter(0, "runnable", 8453888, 0, 1, Ip.iptoa(0), Ip.iptoa(0), "(null)", 0);
```

Paramètres en sortie

Result=1 si ok sinon nil

Exemple

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

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

zz=LPAV_OBJ_ANALOGIC_METER_01;

k=0;
for i=0,20 do
    lpav.Insert(10*i,1,9,9,zz);
    if (k==0) then lpav.SetRef(1,"r"); k=1; else lpav.SetRef(1,"s"); k=0; end
    lpav.SetBrush(0,255,0);
```

```

        lpav.SetFontName("Mistral");
        lpav.SetName(i);
lpav.SetClipart( lp.GetPath().."/card/vumeters/vumeter8.emf");

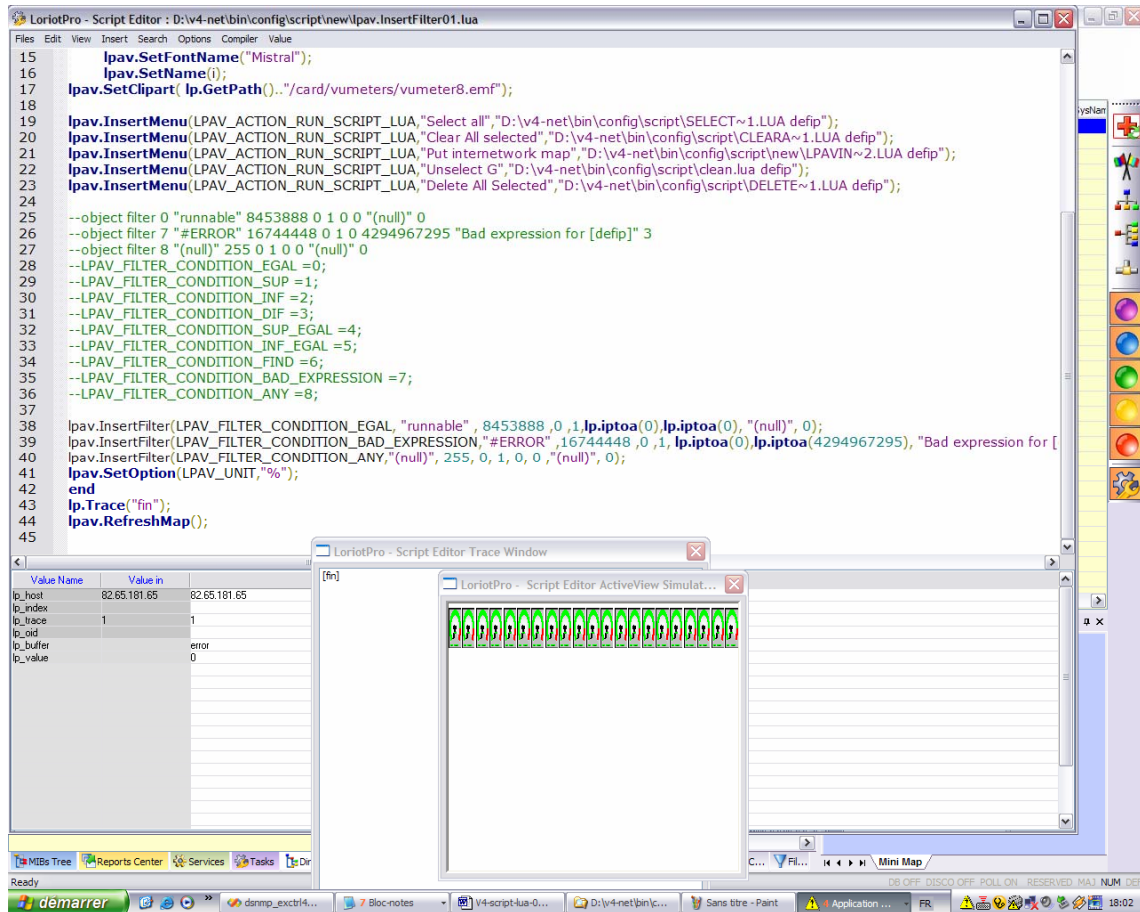
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Select all","D:\v4-
net\bin\config\script\SELECT~1.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Clear All selected","D:\v4-
net\bin\config\script\CLEARA~1.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Put internetwork map","D:\v4-
net\bin\config\script\new\LPAVIN~2.LUA defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Unselect G","D:\v4-
net\bin\config\script\clean.lua defip");
lpav.InsertMenu(LPAV_ACTION_RUN_SCRIPT_LUA,"Delete All Selected","D:\v4-
net\bin\config\script\DELETE~1.LUA defip");

--object filter 0 "runnable" 8453888 0 1 0 0 "(null)" 0
--object filter 7 "#ERROR" 16744448 0 1 0 4294967295 "Bad expression for [defip]" 3
--object filter 8 "(null)" 255 0 1 0 0 "(null)" 0
--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;

lpav.InsertFilter(LPAV_FILTER_CONDITION_EGAL, "runnable" , 8453888 ,0 ,1,lp.iptoa(0),lp.iptoa(0),
"(null)", 0);
lpav.InsertFilter(LPAV_FILTER_CONDITION_BAD_EXPRESSION,"#ERROR" ,16744448 ,0 ,1,
lp.iptoa(0),lp.iptoa(4294967295), "Bad expression for [defip]",3);
lpav.InsertFilter(LPAV_FILTER_CONDITION_ANY,"(null)", 255, 0, 1, 0, 0 ,(null)", 0);
lpav.SetOption(LPAV_UNIT,"%");
end
lp.Trace("fin");
lpav.RefreshMap();

```

LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA



```

static const luaL_reg activeviewlib[] = {
    {"ClearMultiSelect", Ipav_ClearMultiSelect},
    {"SetMultiSelect", Ipav_SetMultiSelect},
    {"IsMultiSelectedMode", Ipav_IsMultiSelectedMode},

    {"RequestString", Ipav_RequestString},

    {NULL, NULL}
};

```


LoriotPro (lpw) Librairie

Cette librairie donne accès à certains modules Wizard de LoriotPro. C'est une librairie dynamique « dll » chargeable avec la fonction Ip.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 (planifié) mais uniquement dans des scripts lancés unitairement.

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

Exemple

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

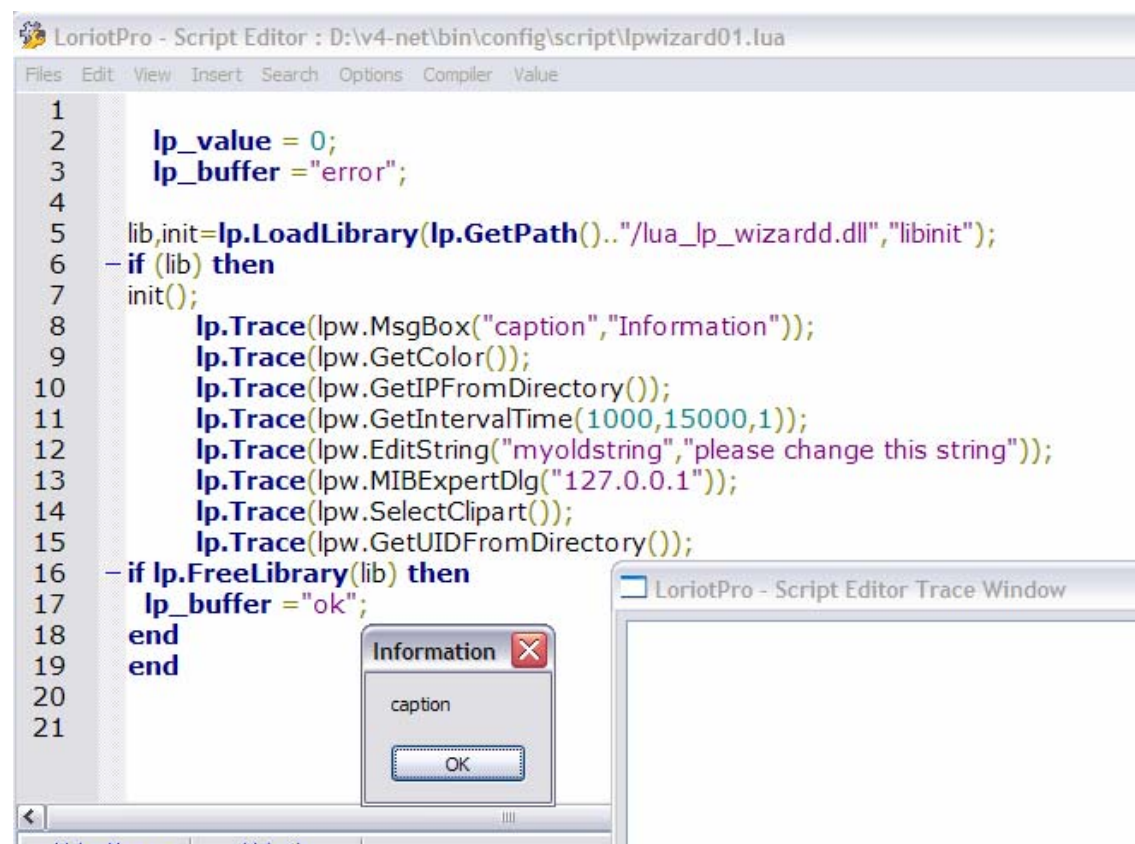
lib,init=Ip.LoadLibrary(Ip.GetPath()."/lua_lp_wizardd.dll","libinit");
if (lib) then
  init();
  Ip.Trace(Ipw.MsgBox("caption","Information"));
  Ip.Trace(Ipw.GetColor());
  Ip.Trace(Ipw.GetIPFromDirectory());
  Ip.Trace(Ipw.GetIntervalTime(1000,15000,1));
  Ip.Trace(Ipw.EditString("myoldstring","please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

lpw.MsgBox

```
lpw.MsgBox("caption","Information");
```

Explication

Cette fonction permet d'afficher une MessageBox.

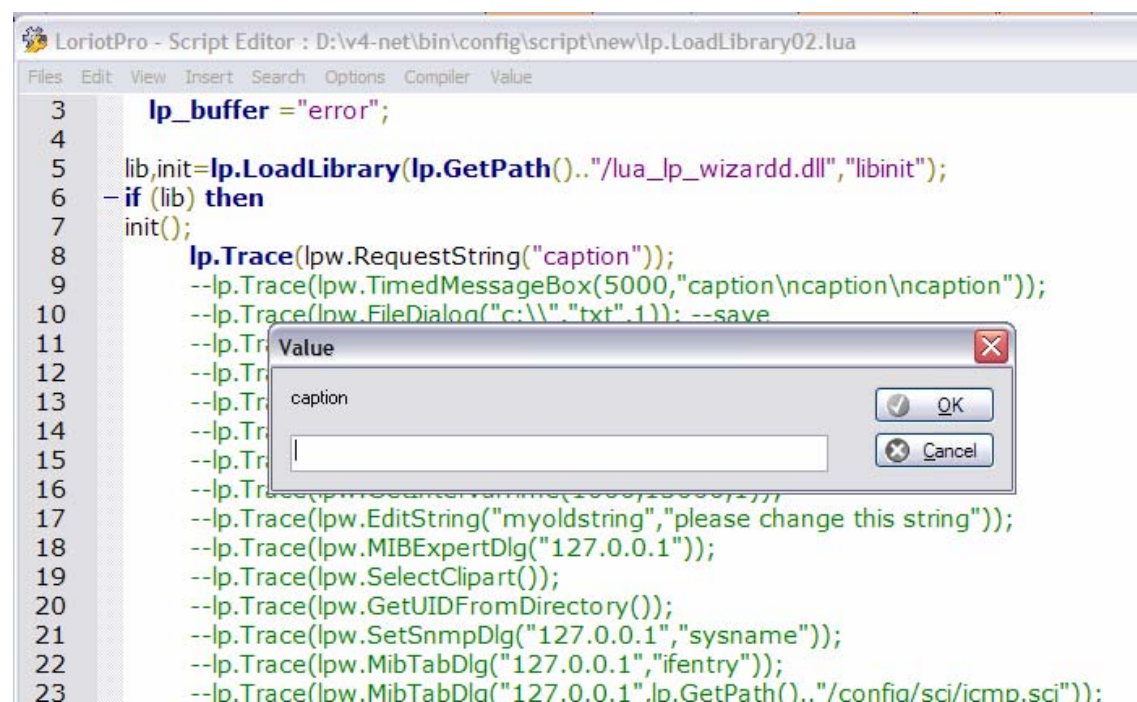


lpw.RequestString

```
'string'=lpw.RequestString ("caption");
```

Explication

Cette fonction permet de collecter une chaîne de caractères saisie par l'utilisateur.



The screenshot shows the 'LoriotPro - Script Editor' window with the following Lua code:

```
3  lp_buffer = "error";
4
5  lib,init=lpw.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6  --if (lib) then
7  init();
8  lp.Trace(lpw.RequestString("caption"));
9  --lp.Trace(lpw.TimedMessageBox(5000,"caption\nncaption\nncaption"));
10 --lp.Trace(lpw.FileDialog("c:\\\".txt".1)); --save
11 --lp.Tr
12 --lp.Tr
13 --lp.Tr
14 --lp.Tr
15 --lp.Tr
16 --lp.Tr
17 --lp.Trace(lpw.EditString("myoldstring","please change this string"));
18 --lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
19 --lp.Trace(lpw.SelectClipart());
20 --lp.Trace(lpw.GetUIDFromDirectory());
21 --lp.Trace(lpw.SetSnmpDlg("127.0.0.1","sysname"));
22 --lp.Trace(lpw.MibTabDlg("127.0.0.1","ifentry"));
23 --lp.Trace(lpw.MibTabDla("127.0.0.1".lp.GetPath().."/confia/sci/icmb.sci"));
```

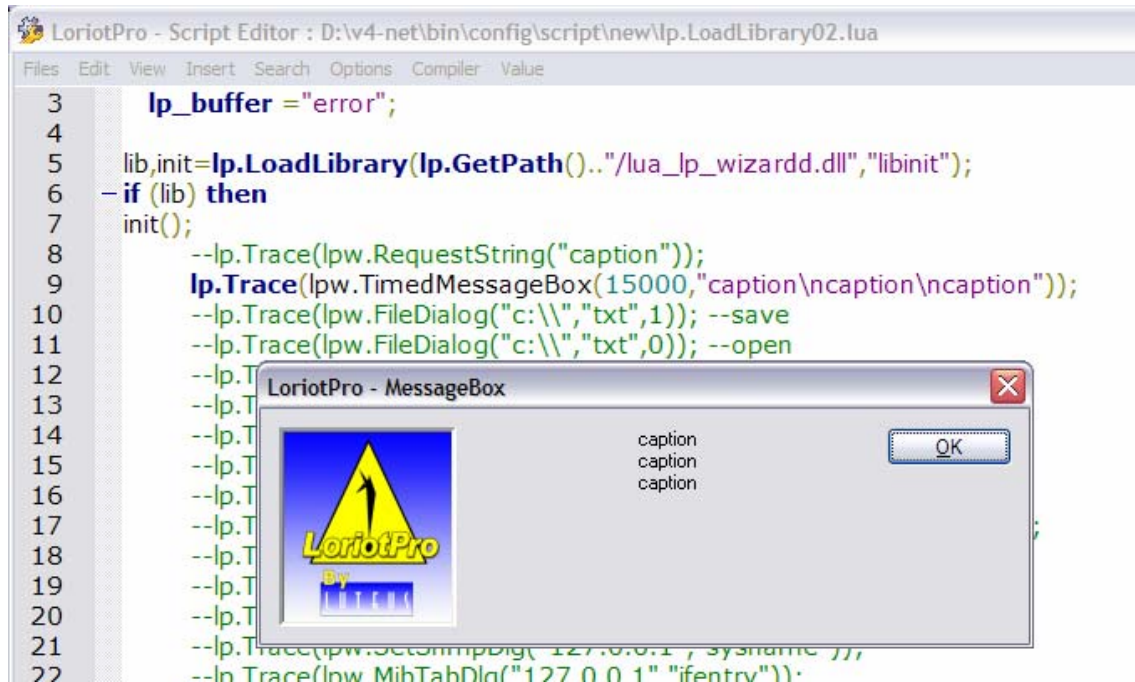
Overlaid on the code is a dialog box titled 'Value'. It has a text input field containing the word 'caption' and two buttons: 'OK' and 'Cancel'.

lpw.TimedMessageBox

```
result=lpw.TimedMessageBox (timeout, "caption1\ncaption2\ncaption3");
```

Explication

Cette fonction permet d'afficher une fenêtre d'information durant un temps donné (timeout).



The screenshot shows the LoriotPro Script Editor with a Lua script and a MessageBox dialog box. The script is as follows:

```
3  lp_buffer = "error";
4
5  lib,init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6  --if (lib) then
7  init();
8      --lp.Trace(lpw.RequestString("caption"));
9      lp.Trace(lpw.TimedMessageBox(15000,"caption\ncaption\ncaption"));
10     --lp.Trace(lpw.FileDialog("c:\\","txt",1)); --save
11     --lp.Trace(lpw.FileDialog("c:\\","txt",0)); --open
12     --lp.T
13     --lp.T
14     --lp.T
15     --lp.T
16     --lp.T
17     --lp.T
18     --lp.T
19     --lp.T
20     --lp.T
21     --lp.Trace(lpw.SetSnmpDir("127.0.0.1",systemc));
22     --lp.Trace(lpw.MibTabDlg("127.0.0.1" "ifentry"));
```

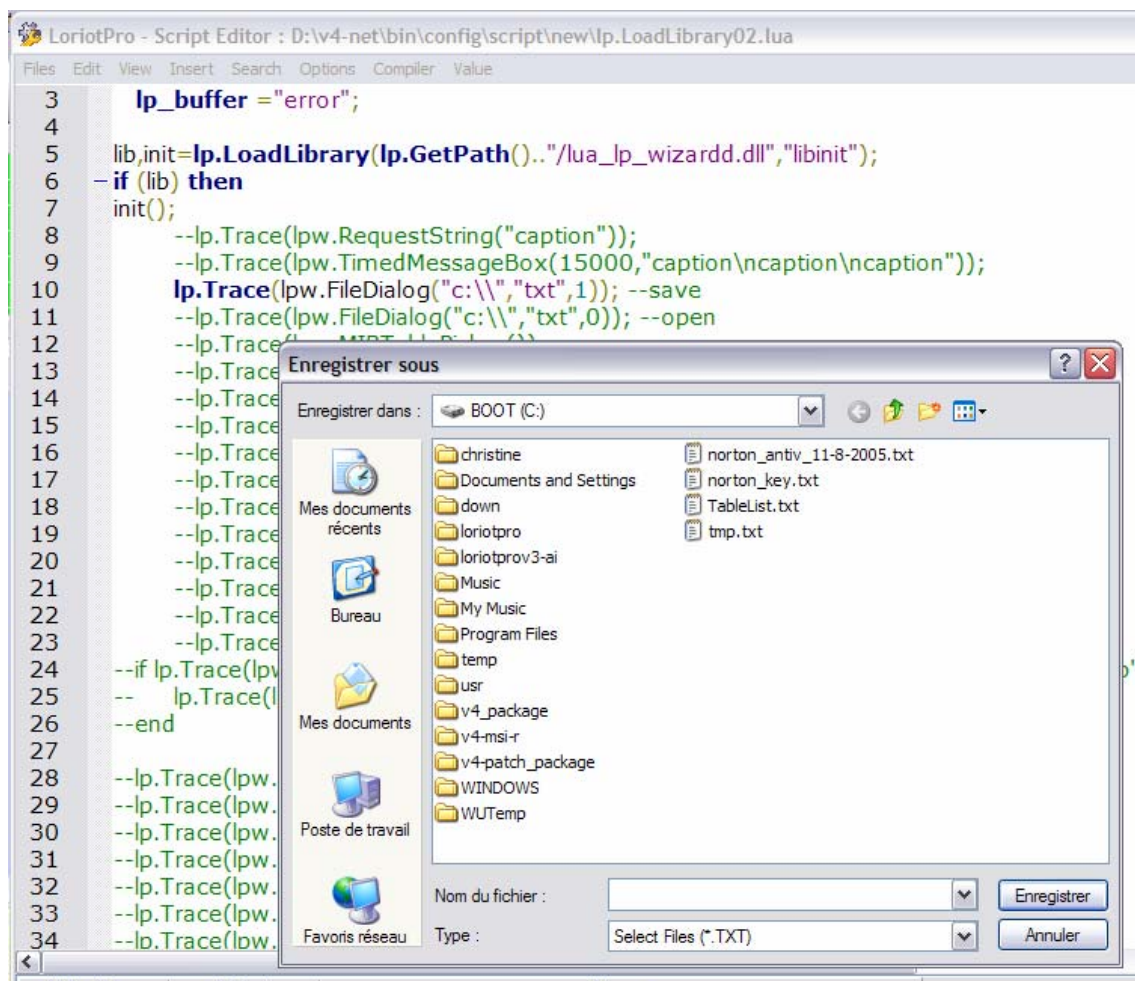
The MessageBox dialog box is titled "LoriotPro - MessageBox" and contains the text "caption", "caption", and "caption" stacked vertically. It has an "OK" button and a red "X" close button in the top right corner. The dialog box also features the LoriotPro logo on the left side.

lpw.FileDialog

'path_file'=lpw.FileDialog ('path','ext',mode); (mode = 0 open =1 save=)

Explication

Cette fonction permet d'afficher une boite de dialogue de sélection de fichier.



Ipw.MIBTablePicker

```
'table_entry'=Ipw.MIBTablePicker ();
```

Explication

Cette fonction permet d'afficher une boîte de sélection des tableaux SNMP disponible dans la base de données des MIB.

The screenshot shows the LoriotPro Script Editor with a Lua script and the MIB Table Picker dialog box. The script is as follows:

```

3  Ip_buffer = "error";
4
5  lib_init=Ip.LoadLibrary(Ip.GetPath().."/lua_ip_wizardd.dll", "libinit");
6  if (lib) then
7    init();
8    --Ip.Trace(Ipw.RequestString("caption"));
9    --Ip.Trace(Ipw.TimedMessageBox(15000,"caption\ncaption\ncaption"));
10   --Ip.Trace(Ipw.FileDialog("c:\\", "txt", 1)); --save
11   --Ip.Trace(Ipw.FileDialog("c:\\", "txt", 0)); --open
12   Ip.Trace(Ipw.MIBTablePicker());
13   --Ip.Trace(Ipw.MsgBox("caption", "Information"));
14   --Ip.Trace(Ip
15   --Ip.Trace(Ip
16   --Ip.Trace(Ip
17   --Ip.Trace(Ip
18   --Ip.Trace(Ip
19   --Ip.Trace(Ip
20   --Ip.Trace(Ip
21   --Ip.Trace(Ip
22   --Ip.Trace(Ip
23   --Ip.Trace(Ip
24  --if Ip.Trace(Ipw.R
25  -- Ip.Trace(Ipw
26  --end
27
28  --Ip.Trace(Ipw.Lo
29  --Ip.Trace(Ipw.Lo
30  --Ip.Trace(Ipw.Lo
31  --Ip.Trace(Ipw.Lo
32  --Ip.Trace(Ipw.Lo
33  --Ip.Trace(Ipw.Lo
34  --Ip.Trace(Ipw.Lo

```

The MIB Table Picker dialog box is open, showing a tree view of loaded MIB tables. The selected table is A3COM0017-STACK-CONFIG, and the selected entry is stackConfigEntry. The dialog box also displays the description of the selected entry:

```

description ""
-----
Object Name      : [ stackConfigEntry ]
Children of Object : [ stackConfigTable ]
From MIB Description : [ A3COM0017-STACK-CONFIG ]
Registered in File : [ 3COM0017.MIB ] The File is loaded
-----
Use Sequence Name : [ StackConfigEntry ]
                  : [ stackUnitLocation,INTEGER ]
                  : [ stackUnitAddress,PhysAddress ]
                  : [ stackUnitLastReset,TimeTicks ]

```

At the bottom of the dialog box, the selected entry name 'stackConfigEntry' is displayed in a text field.

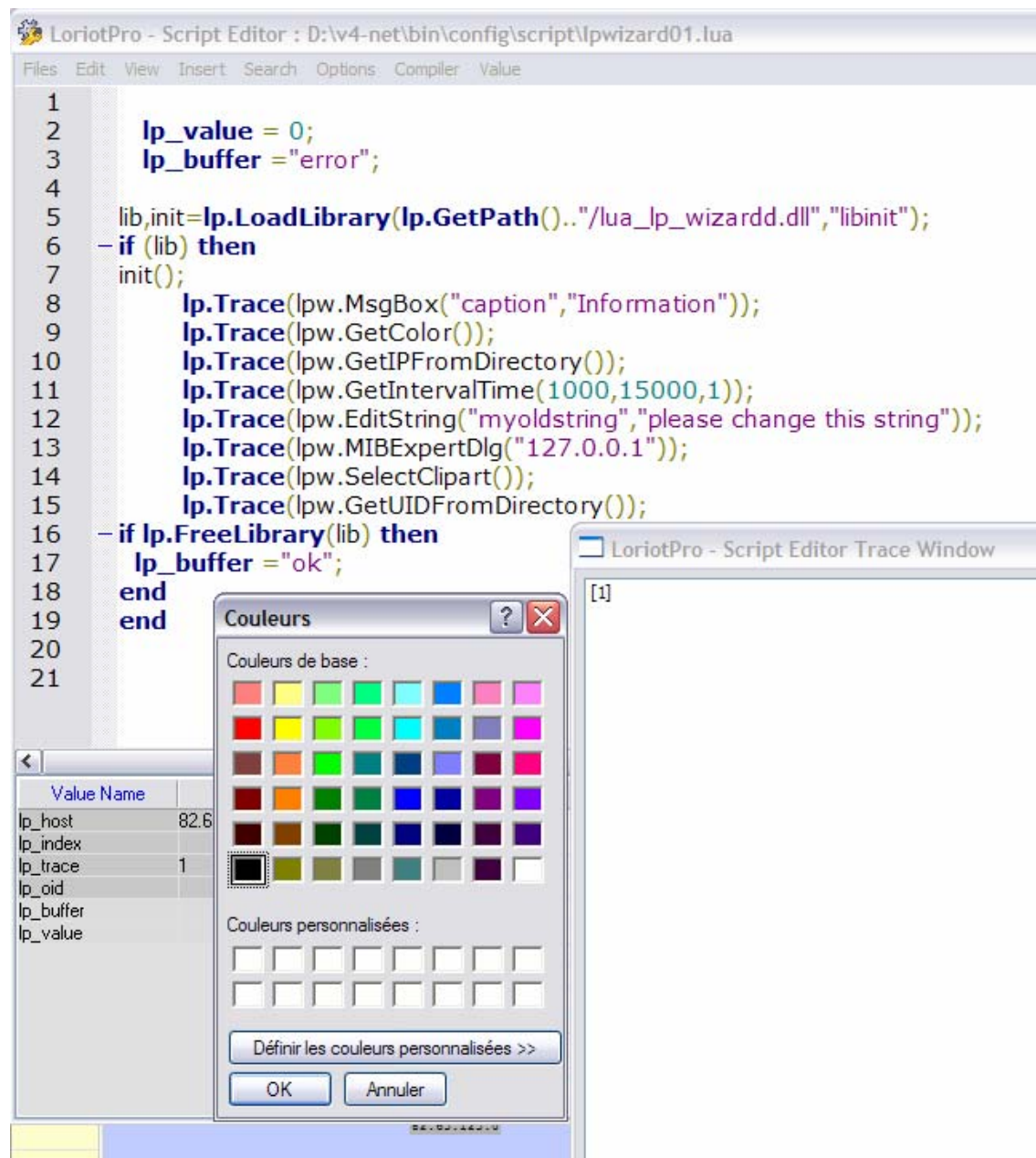
Value Name	Value in
Ip_host	82.65.125.164
Ip_index	
Ip_trace	1
Ip_oid	
Ip_buffer	
Ip_value	

lpw.GetColor

```
r,g,b= lpw.GetColor();
```

Explication

Cette fonction permet d'afficher la boîte de dialogue de sélection de couleur de Microsoft et de récupérer les trois composantes couleurs sélectionnées ou nil.



The screenshot displays the LoriotPro Script Editor with a Lua script and a color selection dialog box. The script is as follows:

```
1
2   lp_value = 0;
3   lp_buffer = "error";
4
5   lib,init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6   -if (lib) then
7     init();
8     lp.Trace(lpw.MsgBox("caption","Information"));
9     lp.Trace(lpw.GetColor());
10    lp.Trace(lpw.GetIPFromDirectory());
11    lp.Trace(lpw.GetIntervalTime(1000,15000,1));
12    lp.Trace(lpw.EditString("myoldstring","please change this string"));
13    lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
14    lp.Trace(lpw.SelectClipart());
15    lp.Trace(lpw.GetUIDFromDirectory());
16   -if lp.FreeLibrary(lib) then
17     lp_buffer = "ok";
18   end
19 end
20
21
```

The dialog box, titled "Couleurs", shows a grid of base colors and a section for custom colors. The "Couleurs de base" section contains a 6x6 grid of color swatches. The "Couleurs personnalisées" section contains a 2x6 grid of empty swatches. Below the grids are buttons for "Définir les couleurs personnalisées >>", "OK", and "Annuler".

The script editor's variable table at the bottom left shows the following values:

Value Name	Value
lp_host	82.6
lp_index	
lp_trace	1
lp_oid	
lp_buffer	
lp_value	

Ipw.GetIPFromDirectory

```
'ip'= Ipw.GetIPFromDirectory();
```

Explication

Cette fonction permet d'afficher la boîte de dialogue de sélection d'une adresse IP de la directory et de retourner une adresse ip sous la forme d'une chaîne de caractères ou nil.

The screenshot displays the LoriotPro Script Editor with a Lua script and its execution results. The script is as follows:

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  lib,init=lp.LoadLibrary(lp.GetPath().."/lua_ip_wizardd.dll","libinit");
6  if (lib) then
7    init();
8    lp.Trace(lpw.MessageBox("caption","Information"));
9    lp.Trace(lpw.GetColor());
10   lp.Trace(lpw.GetIPFromDirectory());
11   lp.Trace(lpw.GetIntervalTime(1000,15000,1));
12   lp.Trace(lpw.EditString("myoldstring","please change this string"));
13   lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
14   lp.Trace(lpw.SelectClipart());
15   lp.Trace(lpw.GetUIDFromDirectory());
16  if lp.FreeLibrary(lib) then
17    lp_buffer = "ok";
18  end
19  end
20
21
```

The execution results are shown in the "LoriotPro - Script Editor Trace Window" and the "Value N" list:

```
[1]
[0][0][0]
```

The "Value N" list contains the following variables:

- lp_host
- lp_index
- lp_trace
- lp_oid
- lp_buffer
- lp_value

The "LoriotPro - Directory Picker" dialog box is open, showing a tree view of the directory structure. The selected host is (0)-LUTEUS02-[192.168.1.105]. The dialog also includes a search bar, OK, Cancel, Collapse All, Expand All, and Help buttons. The text "Just Select an Host in Directory" is displayed at the bottom, and the selected IP address "192.168.1.105" is shown in the input field.

lpw.GetIntervalTime

```
time_out_ms =lpw.GetIntervalTime(time_in_ms,nim_value,control);
```

Explication

Cette fonction permet d'afficher la boite de dialogue de sélection d'un intervalle de temps et de retourner une valeur en millisecondes ou nil.

The screenshot displays the LoriotPro Script Editor with the following Lua code:

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  lib,init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6  -if (lib) then
7  init();
8      lp.Trace(lpw.MessageBox("caption","Information"));
9      lp.Trace(lpw.GetColor());
10     lp.Trace(lpw.GetIPFromDirectory());
11     lp.Trace(lpw.GetIntervalTime(1000,15000,1));
12     lp.Trace(lpw.EditString("myoldstring","please change this string"));
13     lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
14     lp.Trace(lpw.SelectClipart());
15     lp.Trace(lpw.GetUIDFromDirectory());
16  -if lp.FreeLibrary(lib) then
17      lp_buffer = "ok";
18  end
19  end
20
21
```

The 'Interval Time Picker' dialog box is shown with the following details:

- Text field: 1
- Unit selection: Minutes
- Buttons: OK, Cancel, Help

The 'LoriotPro - Script Editor Trace Window' shows the following output:

```
[1]
[0][0][0]
[192.168.1.105]
```

lpw.EditString

```
'new string'=lpw.EditString("my old string","caption : please change this string"));
```

Explication

Cette fonction permet d'afficher la boîte de dialogue d'édition d'une string et de retourner la chaîne de caractères modifiée ou nil.

The screenshot displays the LoriotPro Script Editor with the following Lua code:

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  lib_init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6  -if (lib) then
7    init();
8    lp.Trace(lpw.MessageBox("caption","Information"));
9    lp.Trace(lpw.GetColor());
10   lp.Trace(lpw.GetIPFromDirectory());
11   lp.Trace(lpw.GetIntervalTime(1000,15000,1));
12   lp.Trace(lpw.EditString("myoldstring","please change this string"));
13   lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
14   lp.Trace(lpw.SelectClipart());
15   lp.Trace(lpw.GetUIDFromDirectory());
16  -if lp.FreeLibrary(lib) then
17    lp_buffer = "ok";
18  end
19 end
20
21
```

The Trace Window shows the following output:

```
[1]
[0][0][0]
[192.168.1.105]
[60000]
```

The Value dialog box shows the result of the `lpw.EditString` function call:

Value Name	Value
lp_host	82
lp_index	
lp_trace	1
lp_oid	1

Ipw.MIBExpertDlg

```
'oid'=Ipw.MIBExpertDlg('ip');
```

Explication

Cette fonction permet d'afficher la boîte de dialogue de sélection des objets SNMP et de retourner sous forme d'une chaîne de caractères l'objet SNMP sélectionné ou nil.

The screenshot displays the LoriotPro Script Editor with the following Lua code:

```

1
2  Ip_value = 0;
3  Ip_buffer = "error";
4
5  lib,init=Ip.LoadLibrary(Ip.GetPath().."/lua_ip_wizardd.dll","libinit");
6  -if (lib) then
7    init();
8    Ip.Trace(Ipw.MsgBox("caption","Information"));
9    Ip.Trace(Ipw.GetColor());
10   Ip.Trace(Ipw.GetIPFromDirectory());
11   Ip.Trace(Ipw.GetIntervalTime(1000,15000,1));
12   Ip.Trace(Ipw.EditString("myoldstring","please change this string"));
13   Ip.Trace(Ipw.MIBExpertDlg("127.0.0.1"));
14   Ip.Trace(Ipw.SelectClipart());
15   Ip.Trace(Ipw.GetUIDFromDirectory());
16  -if Ip.FreeLibrary(lib) then
17    Ip_buffer = "ok";
18  end
19  end
20
21

```

The Trace Window shows the output of the script:

```

[1]
[0][0][0]
[192.168.1.105]
[60000]
[my new string]

```

The MIB Object Description Browser Picker window shows the selected object 'ifdescr' and its details:

```

Your Selection: ifdescr
Object Name: [ ifdescr ]
Children of Object: [ ifentry ]
From MIB Description: [ RFC1213-MIB ]
Registered in File: [ kernel.mib ]
The File is loaded
Father Indexed By: [ IfIndex, INTEGER(6) ]

```

The Description window shows the description of the selected object:

```

Description
"A textual string containing information about the interface. This string should include the name of the interface."

```

A dialog box titled "Question" is shown, asking: "Scan Host [LoriotPro] IP [127.0.0.1] for object [ifdescr]". The buttons are "Oui" and "Non".


```
lp_buffer = "ok";  
lib,init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");  
- if (lib) then  
  init();  
  lp.Trace(lpw.MessageBox("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());  
- if lp.FreeLibrary(lib) then  
  lp_buffer = "ok";  
end  
end
```

LoriotPro - MIB Object Description Browser Picker for

LoriotPro - Script Editor Trace Win

```
[1]  
[0][0][0]  
[192.168.1.105]  
[60000]  
[my new string]
```

Mib Row [rfc1213-mib:ifdescr] for 127.0.0.1/LoriotPro

Please select an indexed Snmp object :

Name	Value
ifdescr.65539	VIA Rhine III Fast Ethernet Adapter[00]
ifdescr.1	MS TCP Loopback interface[00]
ifdescr.0	Nul

lpw.SelectClipart

```
'file_path'=lpw.SelectClipart();
```

Explication

Cette fonction permet d'afficher la boîte de sélection des cliparts et de retourner le chemin « path » sur le clipart sélectionné ou nil.

The screenshot displays the LoriotPro Script Editor with a Lua script and a ClipArt File Selector dialog box. The script includes various functions like `lpw.LoadLibrary`, `lpw.Trace`, `lpw.MsgBox`, `lpw.GetColor`, `lpw.GetIPFromDirectory`, `lpw.GetIntervalTime`, `lpw.EditString`, `lpw.MIBExpertDlg`, `lpw.SelectClipart`, and `lpw.GetUIDFromDirectory`. The dialog box shows a list of files and a preview of a selected image.

```
1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  lib_init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6  -if (lib) then
7    init();
8    lp.Trace(lpw.MsgBox("caption","Information"));
9    lp.Trace(lpw.GetColor());
10   lp.Trace(lpw.GetIPFromDirectory());
11   lp.Trace(lpw.GetIntervalTime(1000,15000,1));
12   lp.Trace(lpw.EditString("myoldstring","please change this string"));
13   lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
14   lp.Trace(lpw.SelectClipart());
15   lp.Trace(lpw.GetUIDFromDirectory());
16  -if lp.FreeLibrary(lib) then
17    lp_buffer = "ok";
18  end
19 end
20
21
```

Value Name	Value in
lp_host	82.65.125.164
lp_index	
lp_trace	1
lp_oid	
lp_buffer	
lp_value	

Files

- ..
- architecture 0971.emf
- car001.emf
- car002.emf
- car071.emf
- commerce électronique 0620.emf
- france.emf
- loriot-04.emf
- world.emf

Preview

car002.emf

Load

Cancel

lpw.GetUIDFromDirectory

```
'uid'=lpw.GetUIDFromDirectory();
```

Explication

Cette fonction permet d'afficher la boîte de sélection d'un UID (container) de la Directory et de retourner l'UID sélectionné sous la forme d'une chaîne de caractères ou nil.

The screenshot displays the LoriotPro Script Editor with the following Lua code:

```

1
2  lp_value = 0;
3  lp_buffer = "error";
4
5  lib,init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizardd.dll","libinit");
6  - if (lib) then
7    init();
8    lp.Trace(lpw.MessageBox("caption","Information"));
9    lp.Trace(lpw.GetColor());
10   lp.Trace(lpw.GetIPFromDirectory());
11   lp.Trace(lpw.GetIntervalTime(1000,15000,1));
12   lp.Trace(lpw.EditString("myoldstring","please change this string"));
13   lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
14   lp.Trace(lpw.SelectClipart());
15   lp.Trace(lpw.GetUIDFromDirectory());
16  - if lp.FreeLibrary(lib) then
17    lp_buffer = "ok";
18  end
19 end
20
21

```

The Trace Window shows the following output:

```

[1]
[0][0][0]
[192.168.1.105]
[60000]
[my new string]
[ifdescr.65539]
[d:/v4-net/bin/card/BACKGR~1/car002.emf]

```

The Directory Picker dialog box shows a tree view of the Directory:

- Directory
 - (0)-LoriotPro-{4804718053621236267}
 - (0)-MyOrganisation-{4804718066506138156}
 - (0)-Local_Network_192.168.1.0-{4804718130930647597}
 - (0)-82.65.124.0->SAR_AD_SL_fast-{4804718143815549491}
 - (0)-LUTEUS02-{4804718130930647602}
 - (0)-C2500-11-7-{4807993773638287377}
 - (0)-82.65.125.0->SAR_AD_SL_fast-{4807993756458418190}

The selected object is (0)-LUTEUS02-{4804718130930647602}.

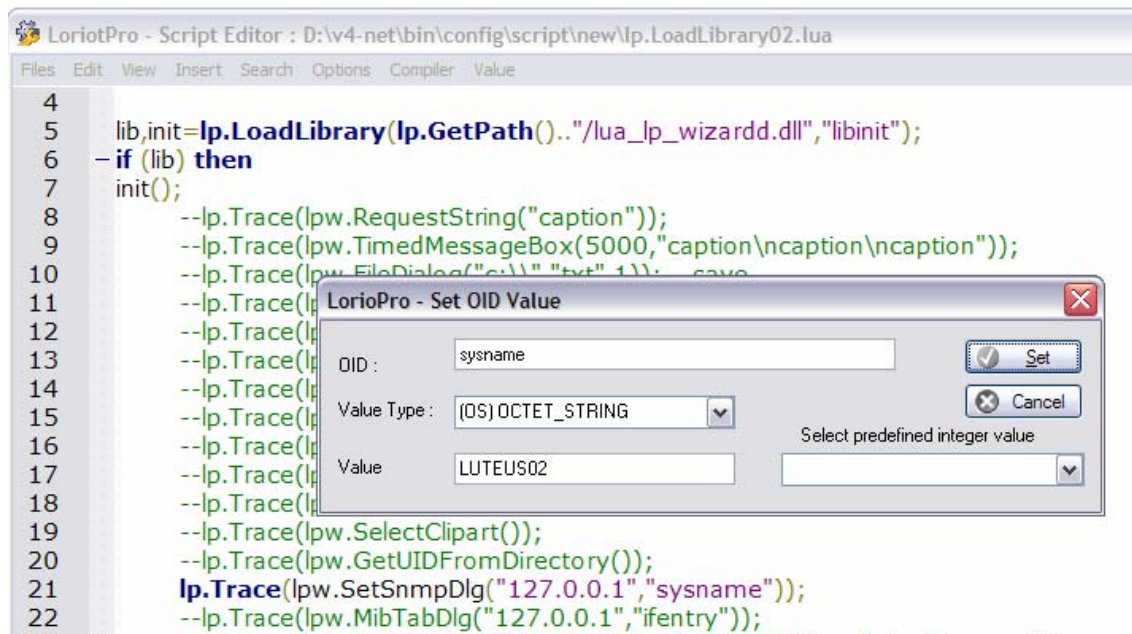
Value Name	Value in	Value out
lp_host	82.65.125.164	82.65.1
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		ok
lp_value		0

lpw.SetSnmDlG

```
result=lpw.SetSnmDlG('ip','OID');
```

Explication

Cette fonction permet d'afficher la boite de dialogue associée à la fonction SNMP SET.

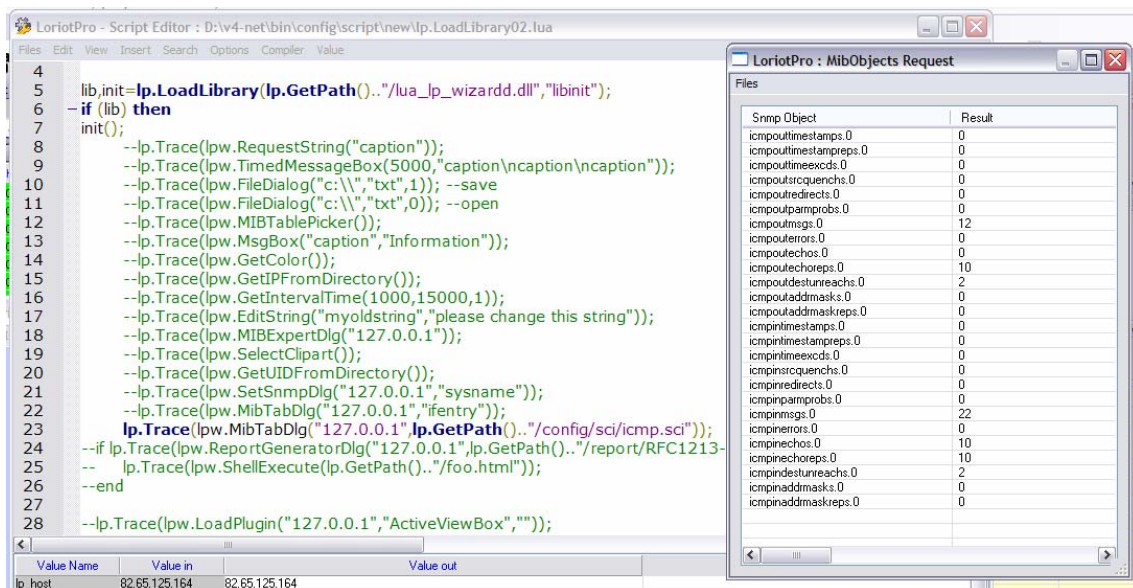
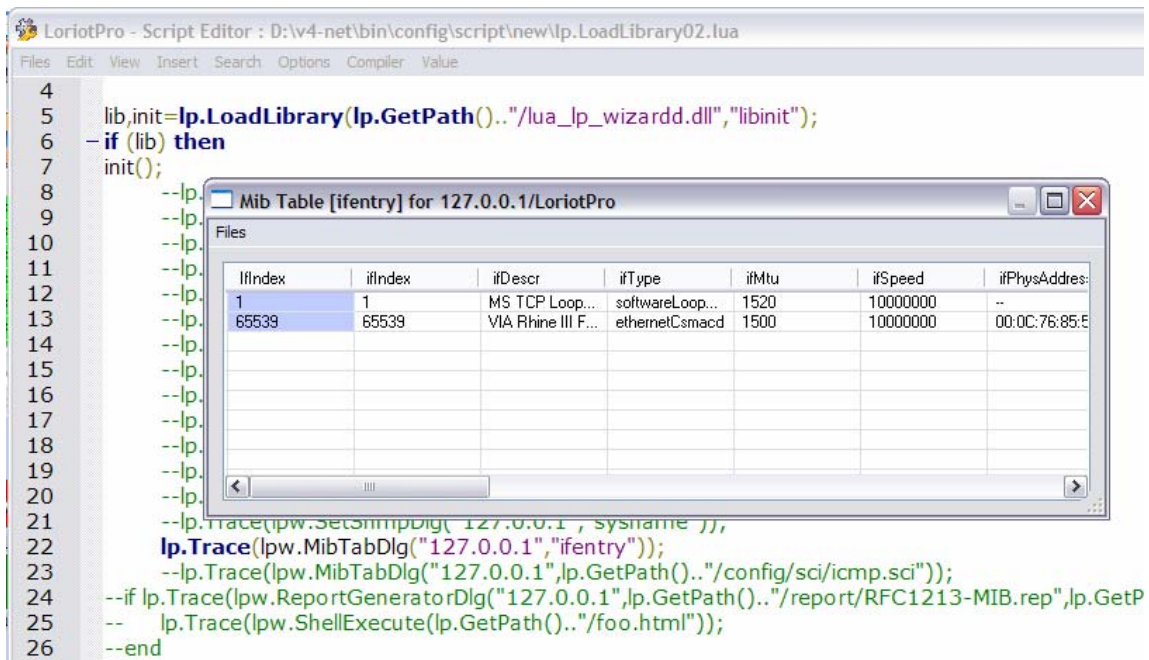


lpw.MibTabDlg

```
result=lpw.MibTabDlg ('ip', 'table_entry_name');
result=lpw.MibTabDlg ('ip', 'sci_file_name');
```

Explication

Cette fonction permet d'afficher la boîte de dialogue permettant l'affichage d'un tableau SNMP.



Ipw.ShellExecute

```
result=Ipw.ShellExecute ('file.html');
```

Explication

Cette fonction permet d'afficher une url dans une fenêtre IE.

lpw.ReportGeneratorDlg

```
result=lpw.ReportGeneratorDlg ('ip','report_file_in.rep','output_file.html');
```

Explication

Cette fonction permet d'afficher la boîte de dialogue de génération de rapports.

```

18      --lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
19      --lp.Trace(lpw.SelectClipart());
20      --lp.Trace(lpw.GetUIDFromDirectory());
21      --lp.Trace(lpw.SetSnmpDlg("127.0.0.1","sysname"));
22      --lp.Trace(lpw.MibTabDlg("127.0.0.1","ifentry"));
23      --lp.Trace(lpw.MibTabDlg("127.0.0.1",lp.GetPath().."/config/sci/icmp.sci"));
24  --if lp.Trace(lpw.ReportGeneratorDlg("127.0.0.1",lp.GetPath().."/report/RFC1213-MIB.rep",lp.GetPath().."/foo.html"));
25      lp.Trace(lpw.ShellExecute(lp.GetPath().."/foo.html"));
26  end
27
28  --lp.Trace(lpw.LoadPlug
29  --lp.Trace(lpw.LoadPlug
30  --lp.Trace(lpw.LoadPlug
31  --lp.Trace(lpw.LoadPlug
32  --lp.Trace(lpw.LoadPlug
33  --lp.Trace(lpw.LoadPlug
34  --lp.Trace(lpw.LoadPlug
35  --lp.Trace(lpw.LoadPlug

```

LoriotPro Rfc1213.mib report script file for : LoriotPro - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente

Rechercher Favoris

Adresse D:\v4-net\bin\foo.html

LoriotPro Report Script, Generated File : from MIB file : Rfc1213.mib

Host Name : *LoriotPro*
Host : **127.0.0.1**
Generated at : *Sun Dec 04 11:33:00 2005*

MIB Definitions of RFC1213-MIB

Group : **system**

Name	sysDescr
Value	Hardware: x86 Family 15 Model 2 Stepping 7 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.1 (Build 2600 Multiprocessor Free)
Description	A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters.

Tree Reports Center Services Tas

Terminé Poste de travail

lpw.LoadPlugin

```
result=lpw.LoadPlugin('ip', 'plugin_name.lp');
```

Explication

Cette fonction permet de lancer certain plugin LoriotPro de type direct (*.lp).

```

27
28 lpw.Trace(lpw.LoadPlugin("127.0.0.1", "ActiveViewBox", ""));
29 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "BulkPollingConfiguration", ""));
30 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "ciscoconfig", ""));
31 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "CiscoISDNStats", ""));
32 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "CollectorSyslogMessages", ""));
33 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "EventsFilterStatistic", ""));
34 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "HostEventsBrowser", ""));
35 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "Re
36 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "Sc
37 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "Re
38 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "Se
39 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "Us
40 --lp.Trace(lpw.LoadPlugin("127.0.0.1", "Whois", ""));
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

--lp.Trace(lpw.LoadPlugin("127.0.0.1", "CollectorSyslogMessages", ""));
lpw.Trace(lpw.LoadPlugin("127.0.0.1", "EventsFilterStatistic", ""));
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "HostEventsBrowser", ""));
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "Re
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "Sc
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "Re
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "Se
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "Us
--lp.Trace(lpw.LoadPlugin("127.0.0.1", "Whois", ""));

```

For Event Number	Total Matched	Matched Since Last Clear Counter	Last Filtered At	
00056000				Event 56000 : test smtp subject toto
4294967295				Event 4294967295 : Match All Event
4294967295	1	1 (il 4 event in 10s)	Sun Dec 04 11:18:35 2005	Filter [4735321681449451638] For [0.0.0.0/0.0.0.0] Action : Wave [D:
4294967295	2	2 (il 5 event in 60s)	Sun Dec 04 11:18:35 2005	Filter [4728627111630209138] For [0.0.0.0/0.0.0.0] Action : Wave [:\
4294967295	2	2 (il 5 event in 60s)	Sun Dec 04 11:18:35 2005	Filter [4728627111630209137] For [0.0.0.0/0.0.0.0] Action : Wave [:\
00000789				Event 789 : Comment ca va bien
00000250				Event 250 : Custom event, no defined in events.txt file
00000214				Event 214 : SnmpV3 Error
00000101				Event 101 : Host go down
00000101	7	7	Sun Dec 04 11:18:35 2005	Filter [4721486986198646888] For [0.0.0.0/0.0.0.0] Action : Wave [C:
00000101	7	7	Sun Dec 04 11:18:35 2005	Filter [4721486986198646835] For [0.0.0.0/0.0.0.0] Action : Wave [C:
00000100				Event 100 : Host go up
00000100	1	1	Sun Dec 04 11:14:48 2005	Filter [4721486986198646794] For [0.0.0.0/0.0.0.0] Action : Wave [we

Attention

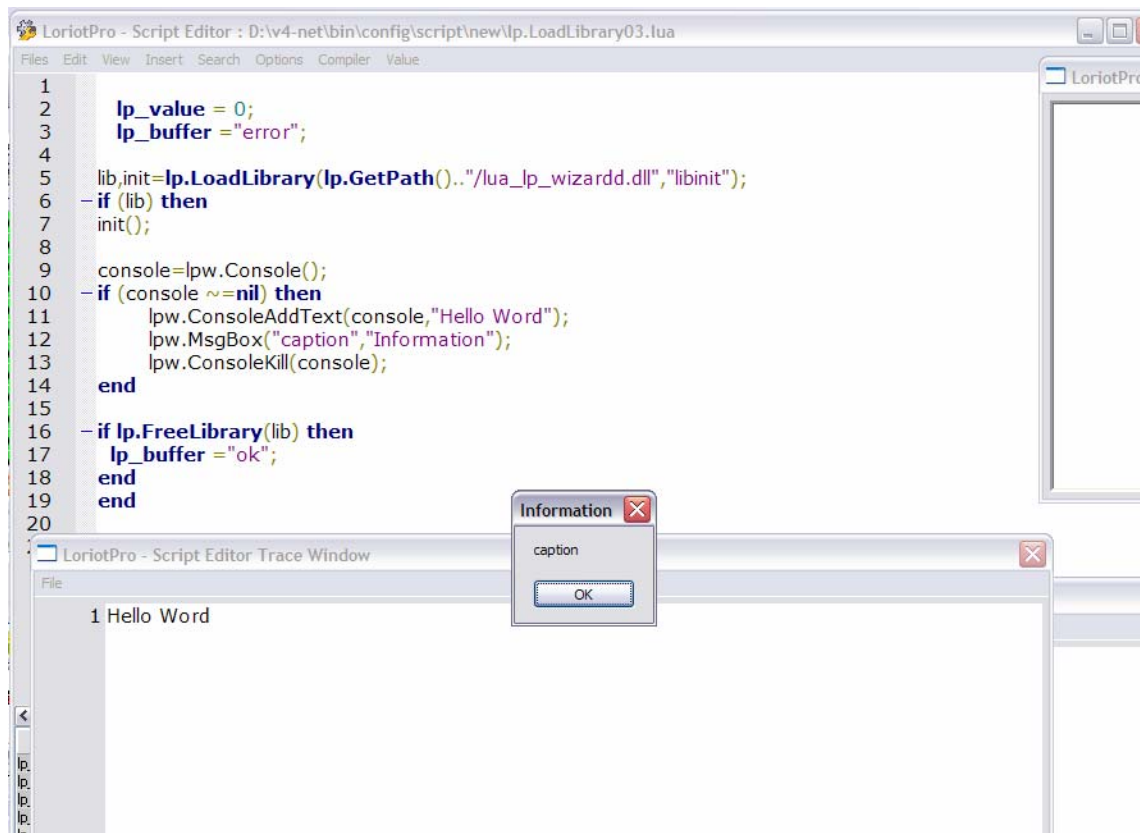
Le plugin *HostProperties.lp* n'est pas supporté.

lpw.Console (beta)

```
console=lpw.Console();
```

Explication

Cette fonction affiche une console pour afficher du texte.



Ipw.ConsoleAddText (beta)

```
result=Ipw.ConsoleAddText(console, « text »);
```

Explication

Cette fonction affiche du texte dans une console.

Ipw.ConsoleKill (beta)

```
result=Ipw.ConsoleKill(console);
```

Explication

Cette fonction détruit la console.

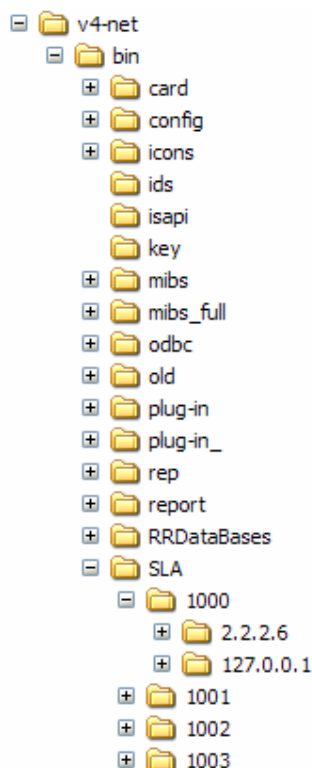
Intégration du SLA dans LUA

Introduction

De base le module de polling (icmp/snmp) intégré à LoriotPro permet de sauvegarder des informations concernant les résultats de polling des hosts. C'est 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 du host interrogé « Poller ». Ces résultats permettent de calculer un SLA sur une période donnée.

Architecture de répertoire

Le logiciel utilise 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.



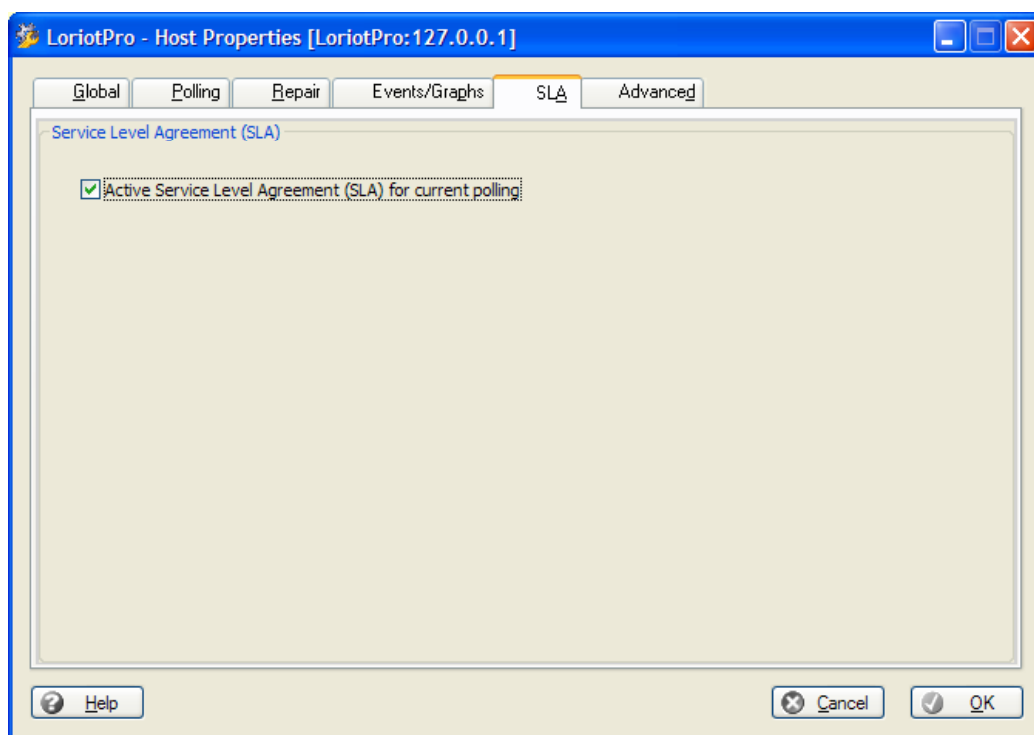
Dans notre capture *V4-net* est le répertoire d'installation du logiciel. Par défaut le logiciel s'installe dans :

C:\Program Files\LUTEUS\LoriotPro V4

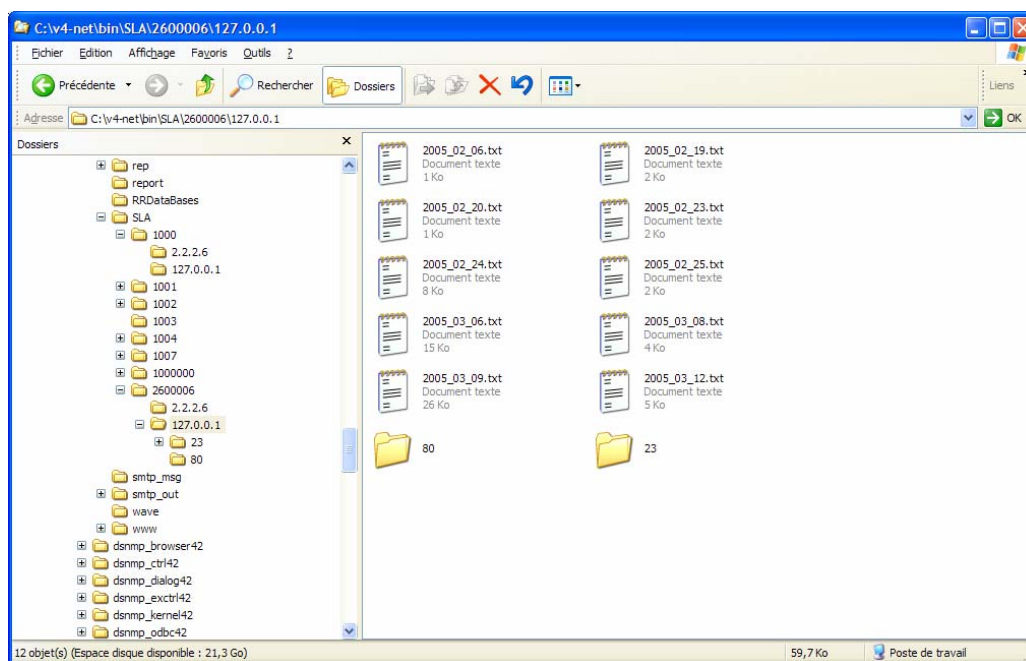
Le répertoire réservé au SLA est dans *\bin\SLA* soit par défaut :

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 host dans le signet SLA.



Les fichiers placés directement dans le répertoire avec l'adresse IP correspondent à la collecte des informations de polling du host. Les répertoires avec l'adresse IP peut contenir d'autre répertoires qui correspondent à des niveaux de SLA plus précis concernant un port UDP TCP ou des URL. La logique de nom des 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 host 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.

Analyse des fichiers

Codage du nom des fichiers

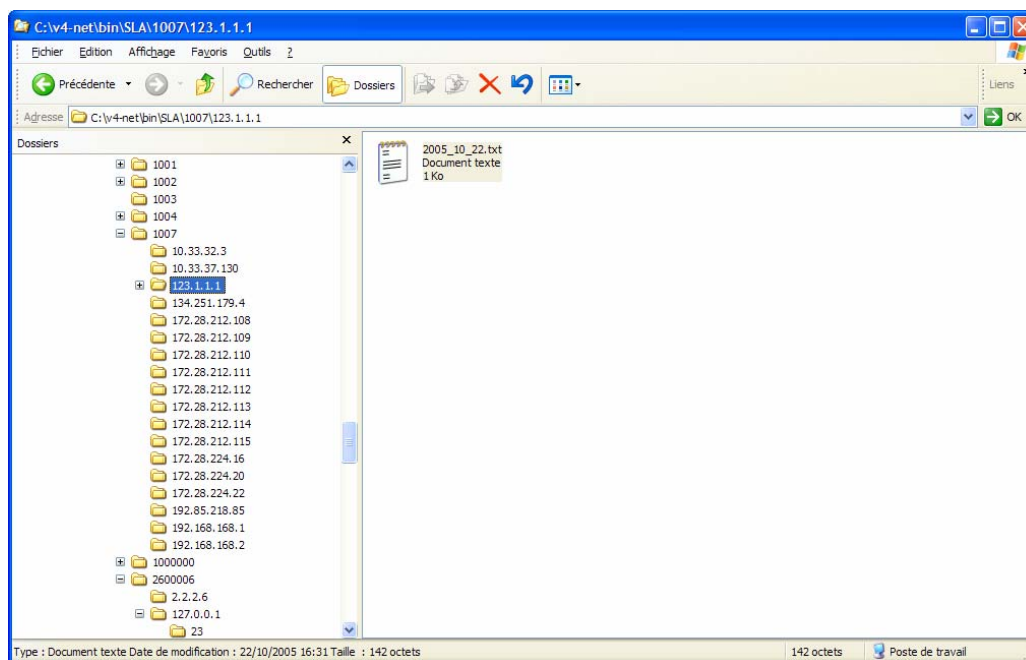
Le nom des fichiers est codé pour une recherche rapide de période. Il y a un fichier généré par jour (365 par ans). Le fichier est au format texte et est lisible par un éditeur standard.

Année_Mois_Jour.txt

Année sur 4 digits

Mois sur 2 digits

Jour sur 2 digits



Dans cette exemple nous avons le fichier de collecte du polling (icmp/snmp) du 22 octobre 2005 pour le host 123.1.1.1 du LoriotPro d'identifiant (ID) 1007.

Contenues des fichiers

Un fichier contient une information générique qui est toujours la même quelque 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
start	indique que le SLA a été activé (ou programme démarré 'mini – bk')
stop	indique que le SLA a été interrompu
stop_polling	le polling est globalement arrêté (valable aussi pour un plugin)
start_polling	le polling est globalement démarré (valable aussi pour un plugin)
Stop_repair	(mini –bk) fin de la période de maintenance
Start_repair	(mini –bk) début de la période de maintenance
Stop_loriot	(mini –bk) le logiciel est stoppé
Start_loriot	(mini –bk) le logiciel est démarré
-1	il n'y a pas eu de réponse de la part du host
35	Une valeur numérique : temps de réponse en millisecondes.

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 cette exemple nous voyons que le SLA a démarré pour ce host en icmp et SNMP le 1129991441, l'intervalle de polling est :

1129991466 - 1129991450 = 16 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 avec la logique d'intervalle 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 cette exemple le host ne répond pas et dans ce cas il y a un double polling icmp (1) et snmp (2).

Si le host 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 le host est en double polling cela veut dire que le host ne répondait plus aux requêtes snmp(2) et que le poller a basculé pour ce host en mode icmp.

Si l'administrateur stoppe le SLA ou le polling, l'information system **stop** est sauvegardé.

```
1129991866;2;-1
```

```
1129991882;1;-1
1129991882;2;-1
1129991895;1;stop
1129991895;2;stop
1129991920;1;start
1129991920;2;start
```

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 est 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 du host pour connaître les vraies valeurs d'intervalle de temps.

Dans le cas du polling d'un host 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.atotp(« 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 envois
ou ss->timeslast correspond au timestamp du dernier reçu
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).

Arrêt du logiciel

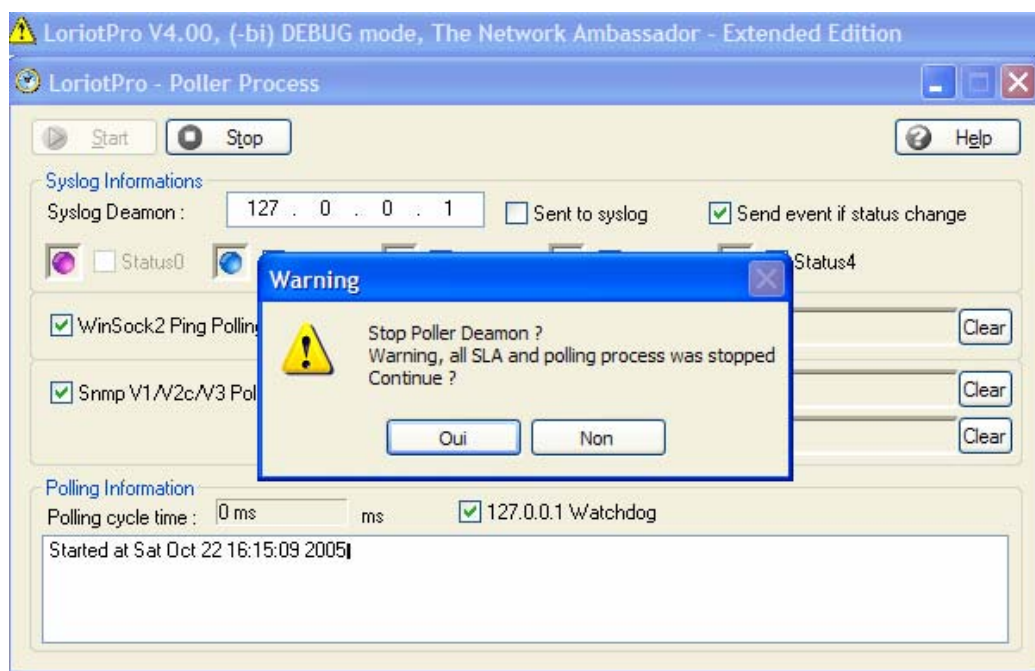
Dans le cas d'un arrêt inopiné du logiciel, il y aura un trou dans le fichier.

```
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 trou entre 1110105908 (**start**) et 1110104432 on en déduit que le LoriotPro a été arrêté.

Arrêt global du polling des hosts

Il est possible de stopper manuellement le polling des hosts.

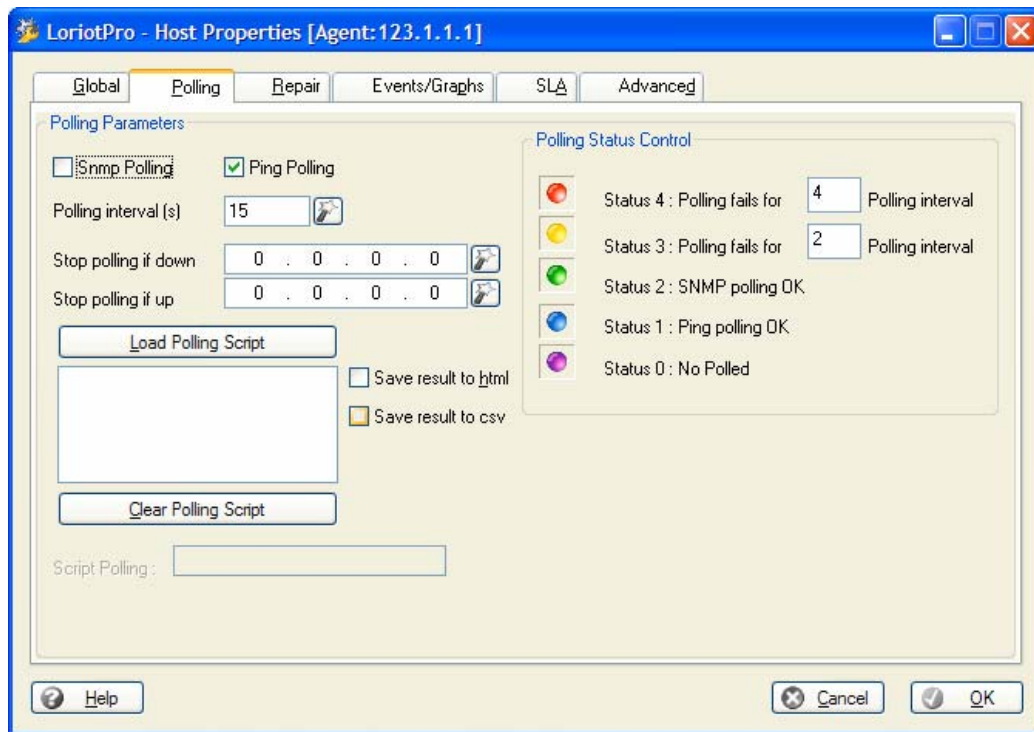


Dans ce cas, il y a une information dans le fichier.

```
1129993499;2;-1  
1129993507;1;stop_polling  
1129993507;2;stop_polling  
1129993511;1;start_polling  
1129993511;2;start_polling  
1129993515;1;-1
```

Arrêt d'un type de polling sur deux

Il est possible de reconfigurer le type de polling d'un host.



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.

SLA Conclusion

Les fichiers de SLA du polling des hosts sont complexes car nous avons un double polling conditionnel (icmp snmp). Pour le « poller » si le snmp répond le icmp n'est pas utilisé. Dans la cas de fichier de SLA du polling d'un host avec uniquement le icmp ou le snmp de validé, l'analyse du fichier est beaucoup plus simple. On peut si on le désire n'analyser que les lignes avec le SNMP (2) qui est prioritaire et considérer que si le host passe en 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 du host)
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
```


LoriotPro (Ipsla) Librairie

Cette librairie donne accès à des fonctions permettant d'exploiter les données SLA associées au host de la directory.

```
V400 b138 SP0-cf 31 mai 2006 :  
ADD lua package: sla librairy
```

La librairie lua_ip_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_ip_slad.dll","libinit");  
else  
lib,init=Ip.LoadLibrary(Ip.GetPath().."/lua_ip_sla.dll","libinit");  
end
```

Ipsla.GetLoriotProIDList

```
number=Ipsla.GetLoriotProIDList('array');
```

Explication

Cette fonction permet de récupérer la liste des racines de collecte SLA par rapport au ID des LoriotPro.

Paramètres en entrée

'array' un tableau « array » de collecte des informations disponible.

Paramètres en sortie

number : le nombre de répertoires de collecte disponibles

array[0] .. array[number-1]

contiennent les ID Loriotpro disponibles.

Ipsla.GetSLAList

```
number=Ipsla.GetSLAList('LoriotProID','array');
```

Explication

Cette fonction fournit la liste des SLA disponibles pour un ID LoriotPro donné (répertoire de collecte).

Paramètres en entrée

'LoriotProID' : un ID (voir le fichier licence.ini) Loriotpro

'array' : un tableau « array » de collecte des SLA disponibles pour ce LoriotProID.

Paramètres en sortie

number : le nombre de SLA disponible

array[0] ... array[number-1] contiennent les SLA (répertoire de collecte)

Ipsla.Compute

```
value=Ipsla.Compute('id','sla_rep',Syear,Smonth,Sday,Eyear,Emonth,Eday,STime,E
Time,RTT_Threshold,Avaibility,Performance,'array')
```

Explication

Cette fonction retourne un tableau contenant les valeurs de SLA pour les paramètres d'entrés fournit

Paramètres en entrée

'id' : un LoriotProID
 'sla_rep' : le répertoire de collecte (ID/SLA)
 Syear : l'année de debut d'analyse du SLA
 Smonth : le mois de debut d'analyse du SLA (1 – 12)
 Sday: le jour de debut d'analyse du SLA (1 – 31)
 Eyear : l'année de fin d'analyse du SLA
 Emonth : le mois de fin d'analyse du SLA
 Eday : le jour de fin d'analyse du SLA
 STime : un timestamp sur le debut de l'analyse du SLA

```
os.time{year=2006,month=5,day=30,hour=0}
```

ETime : un timestamp sur la fin de l'analyse du SLA

```
os.time{year=2006,month=6,day=30,hour=0}
```

RTT_Threshold : le seuil de temps de réponse souhaité pour le calcul

Avaibility : le seuil de disponibilité souhaité

Performance : le seuil de performance souhaité

'array' : Un tableau de collecte des resultants

Valeur de retour dans le tableau	Description
array.ip	L'adresse IP de host concerné par le SLA
array.name	Le nom du host
array.polling_type	Le type de polling du SLA 1 = icmp 2 = snmp
array.periode	En pourcent Le pourcentage de collecte effective sur la période de calcul. Attention dans le cas de host en mode double polling le nombre peut être supérieure à 100%
array.avaibility	Le poucentage de bonne réponse sur la période de calcul (en tenant compte uniquement de la période de résultat)
array.performance	Le poucentage de performance sur la période de calcul (en tenant compte uniquement de la période de résultat) Le paramètre RTT_Threshold est utilisé pour calculer ce pourcentage.

	Pourcentage = poucentage de réponse < à RTT
array.total_collected	Nombre de polling collecté sur la période d'analyse
array.total_waited	Nombre de polling qui aurais du être collecté sur la période d'analyse. Attention dans le cas de host en mode double polling le nombre collecté peut être supérieure au nombre de polling attendu.

Exemple

```

////////// sample
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();

id="1002";
k=lp.sla.GetLoriotProIDList("a");

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

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

if i then

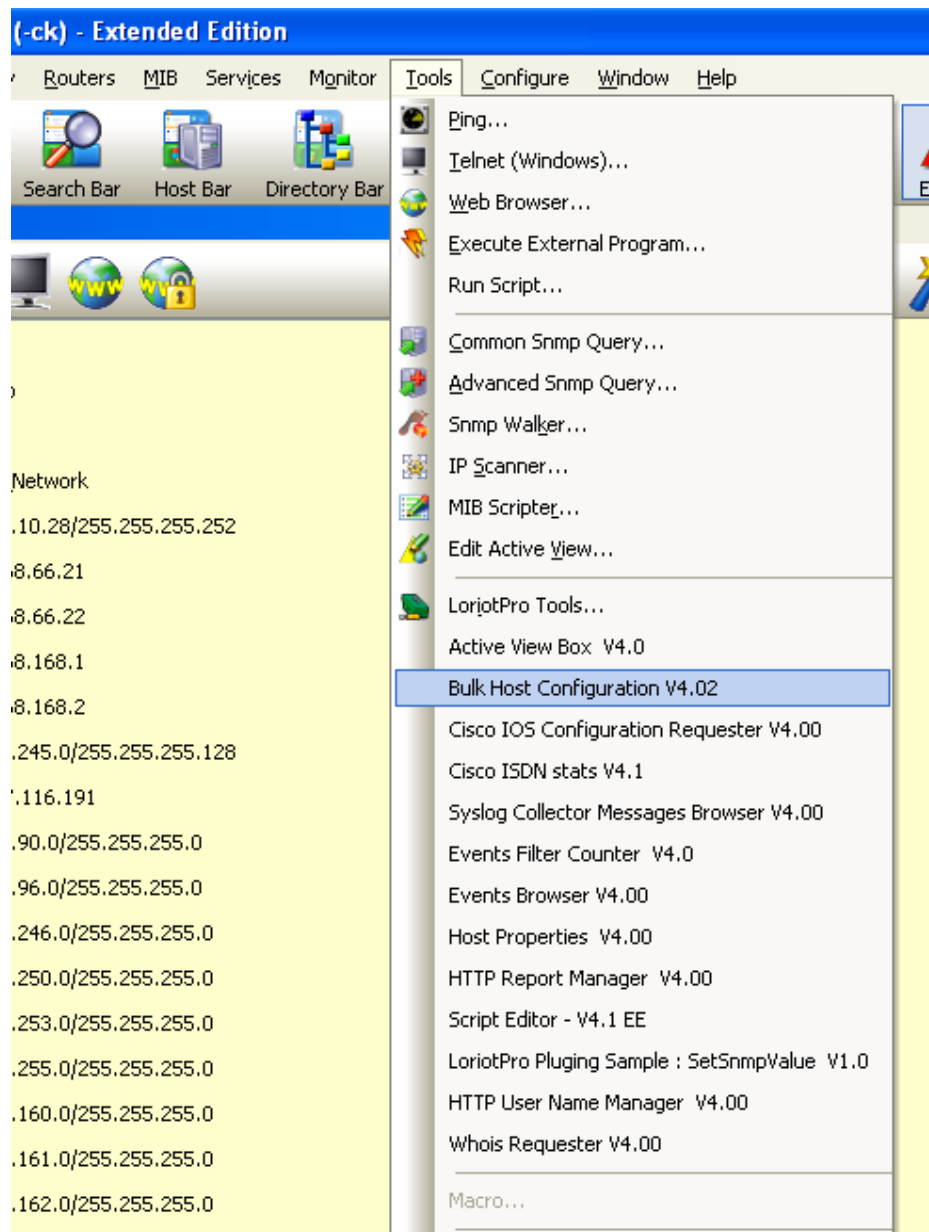
for j=0,i-1 do
lp.Print("\t",aa[j]," SLA \n");

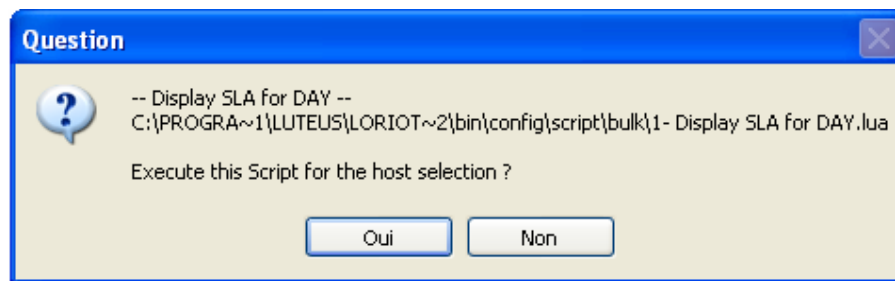
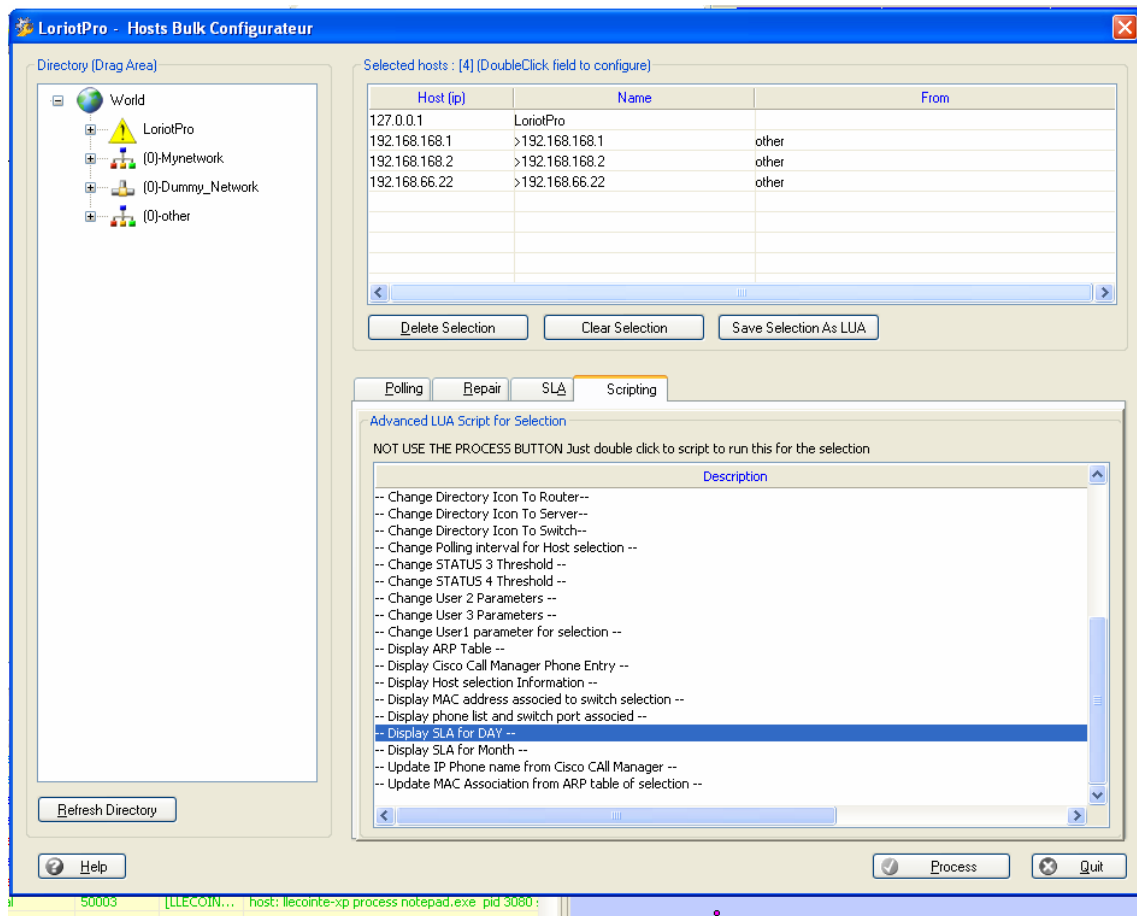
--Compute('id' , 'sla_rep', Syear,Smonth, Sday, Eyear, Emonth, Eday, STime, ETime, RTT_Threshold,
Availability, Performance, 'array')
if
lp.sla.Compute(a[l],aa[j],2005,5,1,2006,6,30,os.time{year=2005,month=5,day=1,hour=0},os.time{year=
2006,month=6,day=30,hour=0},50,90,90,'array') then
lp.Print("\t\tip : ",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\tavaibility : ",array.avaibility,"%\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
end

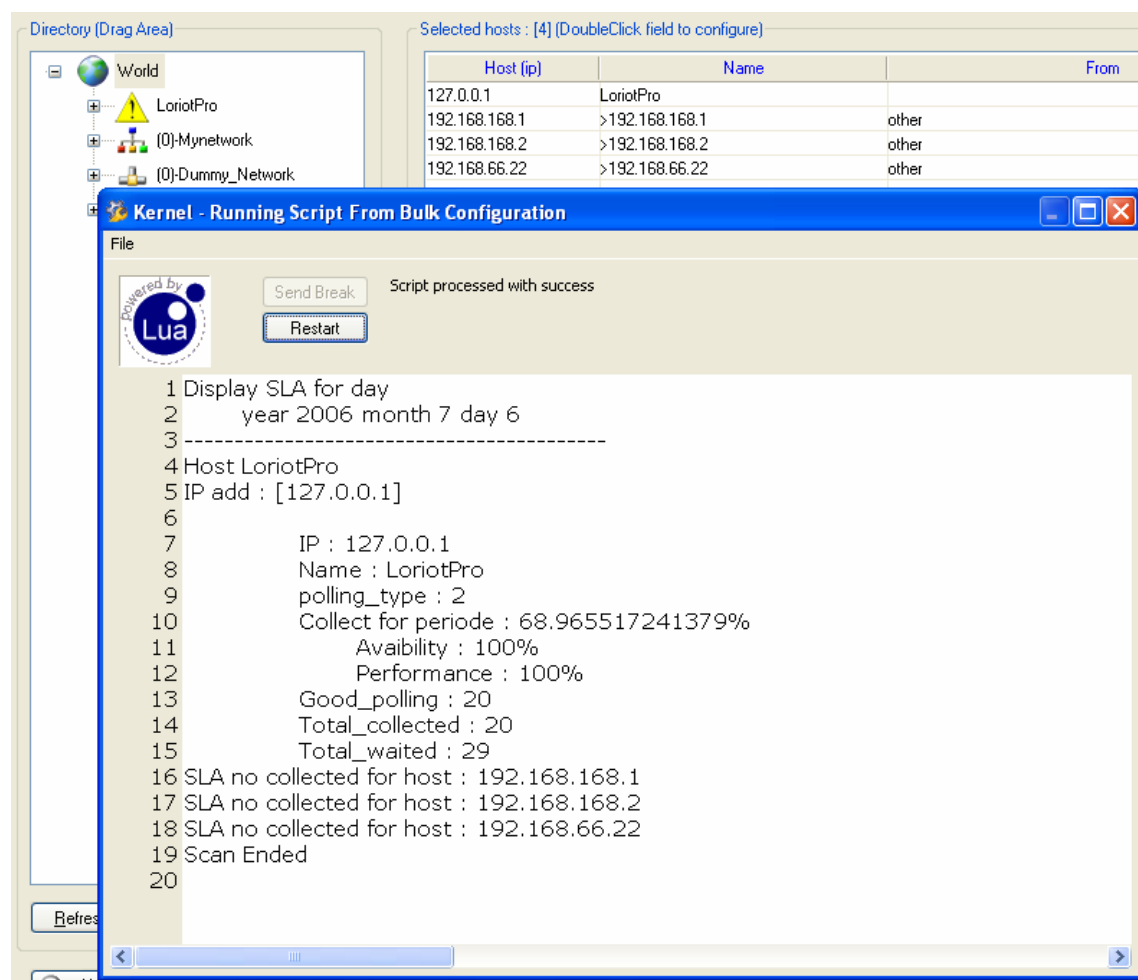
```


LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

Il est possible d'utiliser le module de configuration et de management d'une liste de host pour exploiter les collectes de SLA.







Dans cet exemple, 3 hosts n'ont pas été configurés pour une collecte des polling et le calcul de SLA n'est pas possible. Le host 127.0.0.1 est correctement configuré mais le nombre de polling attendu (total_waited) est supérieur au nombre reçu alors qu'il n'y a pas de perte. Ceci s'explique par le fait que l'intervalle de polling pour le host a été modifié depuis le départ de la collecte et le référentiel de temps a été modifié (temps de polling diminué).

Script utilisé

```

-- Display SLA for DAY --
-- 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";
-- use this to initialise the host selection
dofile(lp.GetPath().."config/script/bulk/selection/LP_Selection.lua")
dofile(lp.GetPath().."config/script/lib-audit/1-audit.lua");
  
```

```

-- Start program
-----
--list the ip host to scan
tabz={};
hostnumber=LP_HostsSelection(tabz);

if hostnumber==0 then error("Not host selected\n") end

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==nil) then error("SLA Lib Not found or not loaded\n") end;

init();
lp.Print("Display SLA for day\n");
temp=os.date("**t",os.time());
--[
temp.year
temp.month
temp.day
temp.hour
temp.min
--]
lp.Print(string.format("\tyear %i month %i day %i\n",temp.year,temp.month,temp.day));

for i=0,table.getn(tabz) do

    info={};
    rep=lp.GetIPInformation(tabz[i],"array");

    if rep then
        if array.sla==1 then

            lp.Print(string.format("-----\nHost %s\nIP add :
[%s]\n\n",array.name,tabz[i]));

            if
lpsla.Compute(100001,tabz[i],temp.year,temp.month,temp.day,temp.year,temp.month,temp.day
,os.time{year=temp.year,month=temp.month,day=temp.day,hour=0}
,os.time{year=temp.year,month=temp.month,day=temp.day,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\tCollect for periode : ",array.periode,"%\n");
                lp.Print("\t\tAvaibility : ",array.avaibility,"%\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
    else

```

```
        Ip.Print("SLA no collected for host : ",array.ip," \n");
        end
    end
end
Ip.Print("Scan Ended\n");
Ip_buffer ="ok";

    end
end
Ip.Print("Scan Ended\n");
Ip_buffer ="ok";
```

Exemple de code sous forme de fonction LUA

Vous trouverez dans ce chapitre des exemples de fonctions pour vous permettre de vous familiariser avec le langage LUA intégré à LoriotPro.

Gestion de périodes

Cette ensemble de fonction permet de déterminer des période de temps

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

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
```

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
```


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
```

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
```

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
```

function LP_ComputeUpTime

Cette fonction convertie la valeur fournit par des objets SNMP tel que sysuptime en chaine de caractères.

```
-----  
function LP_ComputeUpTime(value)  
if value==nil then return "Uptime" end  
value=value/100; -- passe en seconde  
--86400 seconde par jours  
days=value/86400;  
days=string.format("%.0f",days); --sup les valeur après le point  
reste=(value - (days*86400));  
if (reste < 0 ) then return (string.format("%.0f days 0 hours 0 minutes 0 secondes",days)) end  
--1 hours = 3600 seconde  
hours=reste/3600;  
hours=string.format("%.0f",hours); --sup les valeur après le point  
reste=reste - (hours*3600);  
if (reste < 0 ) then return (string.format("%.0f days %.0f hours 0 minutes 0 secondes",days,hours)) end  
--1 minutes = 60 secondes  
minutes=reste/60;  
minutes=string.format("%.0f",minutes); --sup les valeur après le point  
secondes=reste-(minutes*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
```

Analyse de ressource SNMP

Ces Fonctions permette d'analyser des ressources SNMP d'un host, les paramètres en entrés sont l'adresse IP des host un numero d'événement a uncertain level qui sera envoyé si un certain seuil est atteint.

function LP_AuditStorage

Cette fonction permet d'auditer la taille disque d'un host 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["I-".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

end

end
return 1;

end
end
```

function LP_AuditSystem

```
-----  
function LP_AuditSystem(ip,event,level,tab,file)  
number=ip.Gets(ip,"sysdescr,sysobjectid,sysuptime,syscontact,sysname,syslocation,syservices,ipForwarding","a  
a");  
uptime,buffer=ip.Get(ip,"sysuptime");  
ip.Print("Audit system Information for : ",ip," ",number,"\n")  
  
if number~=8 then  
ip.Print("Host no repond or error\n");  
return 0;  
end  
--last=(aa["sysuptime.0"])*1;  
  
ip.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["syservices.0"]  
,aa["ipforwarding.0"]  
));  
  
tab["name"]=aa["sysname.0"];  
tab["objectid"]=aa["sysobjectid.0"];  
tab["services"]=aa["syservices.0"];  
tab["uptime"]=uptime;  
tab["forwarding"]=aa["ipforwarding.0"];  
return 1;  
  
end
```

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

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
```

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,"iproudest,iprouteifindex,iproutemetric1,iproutemetric2,iproutemetric3,iproutemetric4,iproutenexthop,iprouteproto,iprouteage,iproutemask", "a");
if i~=nil then lp.Print("List Routes(",i,")\n") end ;
    for j=0,(i-1) do
if a["iproudest-"..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["iproudest-"..j],a["iproutemask-"..j],a["iprouteproto-"..j]
,a["iproutenexthop-"..j],a["iprouteifindex-"..j],a["iprouteage-"..j]
,a["iprouteproto-"..j],a["iprouteage-"..j],a["iproutemetric1-"..j]
,a["iproutemetric2-"..j],a["iproutemetric3-"..j]));
        end
end
```

function LP_AuditDot3Stat

```

-----
function LP_AuditDot3Stat(ip,index,level,file)
req=string.format("dot3statsalignmenterrors.%s,dot3statsfcerrors.%s,dot3statssinglecollisionframes.%s,dot3stat
smultiplecollisionframes.%s,dot3statssqetesterrors.%s,dot3statsdeferredtransmissions.%s,dot3statslatecollisions.
%s,dot3statsexcessivecollisions.%s,dot3statsinternalmacstransmittererrors.%s,dot3statscarriersenseerrors.%s,dot3st
atsframeoolongs.%s,dot3statsinternalmacreceiveerrors.%s"
,index,index,index,index,index,index,index,index
,index,index,index,index);
--lp.Print(ip," ",req,"\n");
rcv=lp.Gets(ip,req,"aa");

    if (rcv) then
        --lp.Print("\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["dot3statssqetesterrors"..index]~="0" and aa["dot3statssqetesterrors"..index]~=nil then
lp.Print("\t\t\t", "SQE Test Errors(", string.format("%s",aa["dot3statssqetesterrors"..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["dot3statsinternalmacstransmittererrors"..index]~="0" and
aa["dot3statsinternalmacstransmittererrors"..index]~=nil then lp.Print("\t\t\t", "Internal MAC
Transmittererrors(", string.format("%s",aa["dot3statsinternalmacstransmittererrors"..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["dot3statsframeoolongs"..index]~="0" and aa["dot3statsframeoolongs"..index]~=nil then
lp.Print("\t\t\t", "Frame Too Longs(", string.format("%s",aa["dot3statsframeoolongs"..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

```

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 repond\n");
    return 0;
end
i = lp.GetRows(ip,"cpmcpototal5sec,cpmcpototal1min,cpmcpototal5min","a");
if i then lp.Print("List CPU Usage in %\n")
    for j=0,(i-1) do
        result=string.format("cpmcpototal5sec[%s%%] cpmcpototal1min[%s%%]
cpmcpototal5min[%s%%]",a["cpmcpototal5sec-".j],a["cpmcpototal1min-".j],a["cpmcpototal5min-".j]);
        lp.Print("\tindex["..a["I-".j], " ]\t\t",result,"\n");

        if (lp.atof(a["cpmcpototal5sec-".j])>threshold) or (lp.atof(a["cpmcpototal1min-".j])>threshold) or
(lp.atof(a["cpmcpototal5min-".j])>threshold) then
            rep=lp.GetIPInformation(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
            lp.SendEvent( event,level,ip,"255.255.255.255",alarm);
        end
    end
end
end
```

Librairies externe LUA

Avec la fonction « lp.LoadLibairy » il est possible de charger des librairies écrites par des tiers. Si vous chercher 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

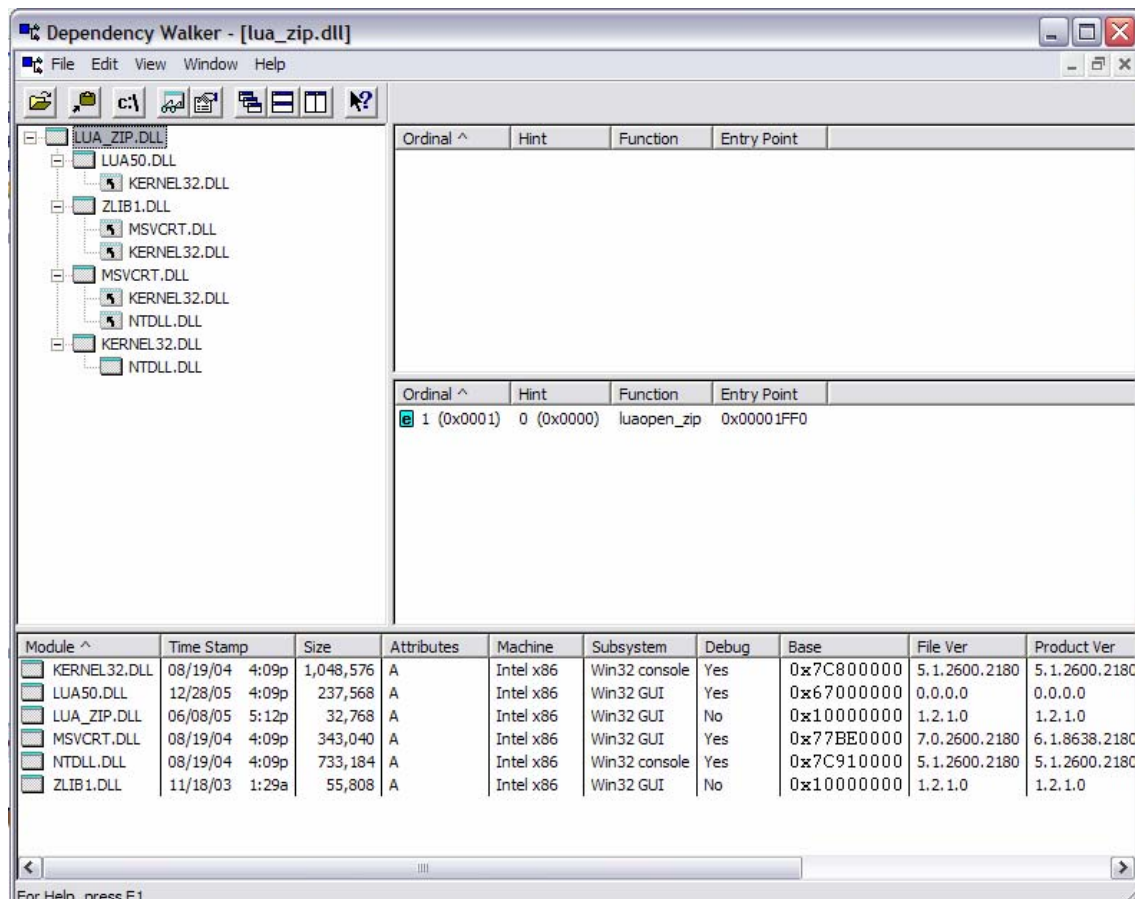
L'usage des librairies externe n'est pas supporté par LUTEUS.

Pour être compatible, les librairies utilisées doivent être compilées en mode dll avec la version 5.02 de LUA. La dll LUA appelé par la librairie doit être lua50.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 chainage des DLL.

Exemple LUA_ZIP

Voici par exemple comment intégrer le package luazip 1.2.1 que vous pouvez trouver sur le site du projet Kepler <http://www.keplerproject.org/> .

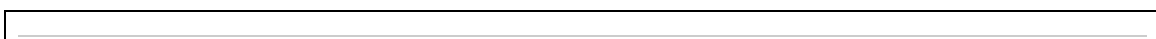
1. Télécharger le fichier luazip-1.2.1-win32.zip.
2. Le fichier luazip-1.2.1-win32.zip contient deux DLL zip.dll et zip1.dll nous renomons zip.dll en lua_zip.dll pour ne pas effacer la dll zip.dll déjà présente dans le répertoire bin du logiciel.
3. Copier les deux dll dans le répertoire bin.
4. Utiliser 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 package contiennent la documentation pour permettre l'utilisation de la librairie.

Extrait de la documentation de lua_zip.



Copyright © 2003-2005 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 [I/O 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 functions 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 functions 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 functions use 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'
- 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 utiliser la fonction **Ip.LoadLibrary** et nom pas require.

Exemple :

```
Ip_value = 0;
Ip_buffer = "error";

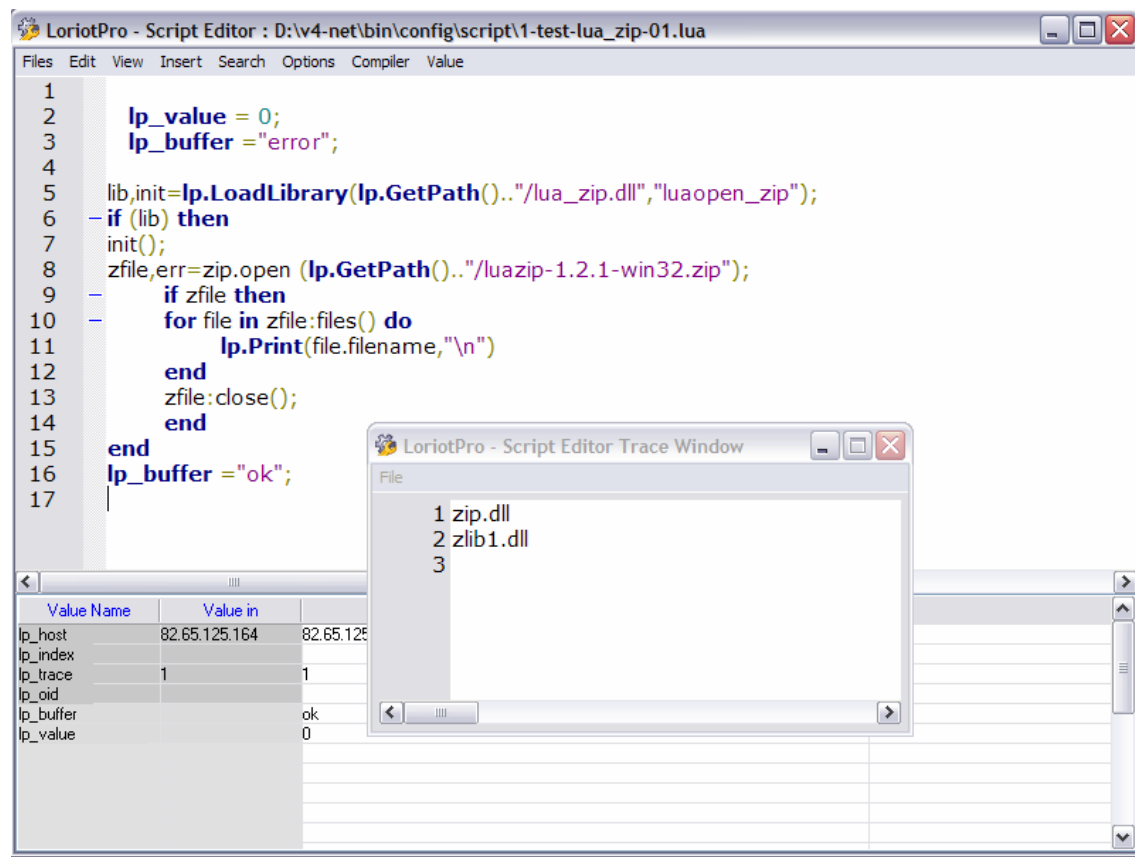
lib,init=Ip.LoadLibrary(Ip.GetPath().."/lua_zip.dll","luaopen_zip");
if (lib) then
init();
```



```

zfile,err=zip.open (lp.GetPath().."/luazip-1.2.1-win32.zip");
  if zfile then
    for file in zfile.files() do
      lp.Print(file.filename,"\n")
    end
    zfile:close();
  end
end
lp_buffer ="ok";

```



Exemple LUA_ODBC

La librairie lua_odbc (<http://www.keplerproject.org/luasql/>) a été testé avec succès pour accéder aux données de la base de données de LoriotPro.

Exemple :

```
-----
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
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_odbc.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();
```

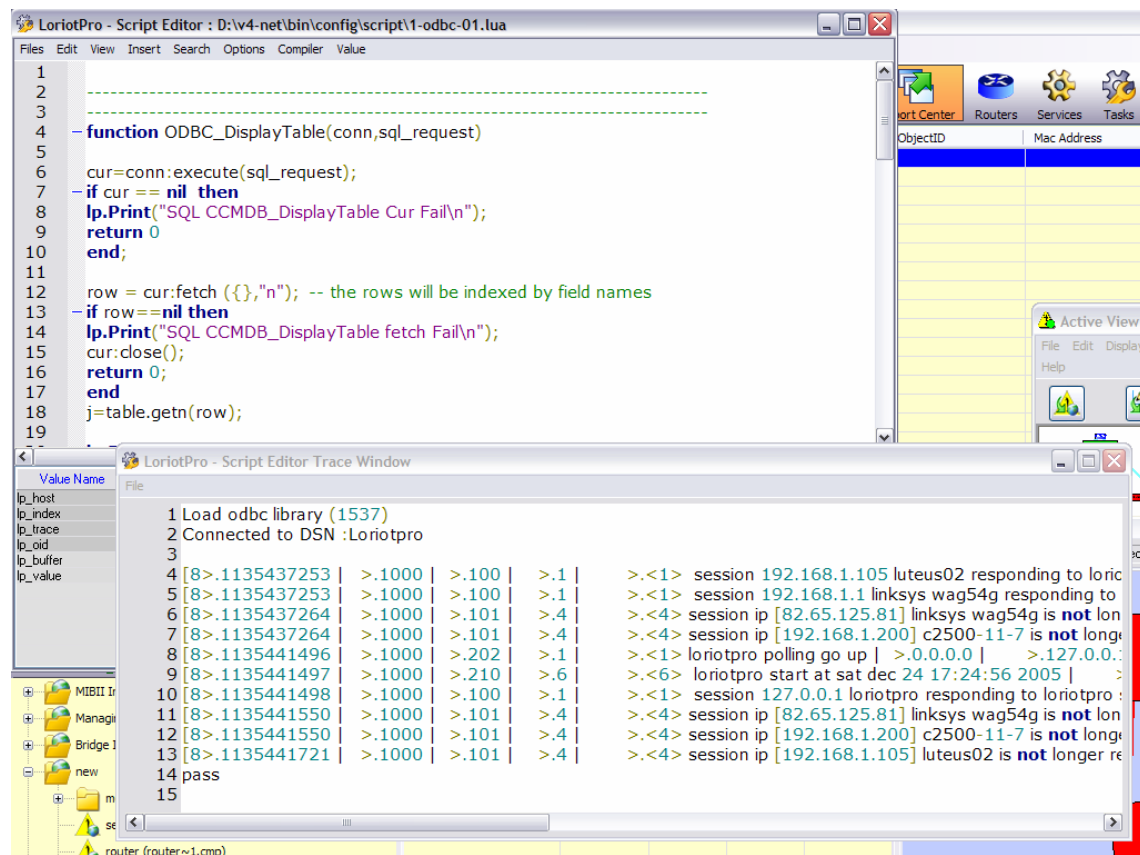
LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

```

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 I_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
end
env:close();
end

cur:close();
conn:close();
env:close();
lp.Print("pass\n");

```



Exemple LUA_COM

L'utilisation de la library luacom <http://www.tecgraf.puc-rio.br/~rcerq/luacom/> est rendue possible uniquement en mode menu direct (ouverture de l'interface microsoft COM). Cette library est extrêmement puissante et permet de réaliser un interface entre LUA et COM.

ATTENTION

L'usage de cette library ainsi que toute les libraries 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.

Exemple d'utilisation de WMI

```

local lib,init= lp.LoadLibrary(lp.GetPath().."/lua_com.dll","luaopen_luacom")

if lib then init();
lp.Print("loacom 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
    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,
end,

```

```

GetFreeMemory = function (self)
    local x
    self.oRefresher:Refresh ()
    for index,item in luacomE.pairs (self.refobjMemory) do
        x = item:AvailableMBytes ()
    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()
        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")

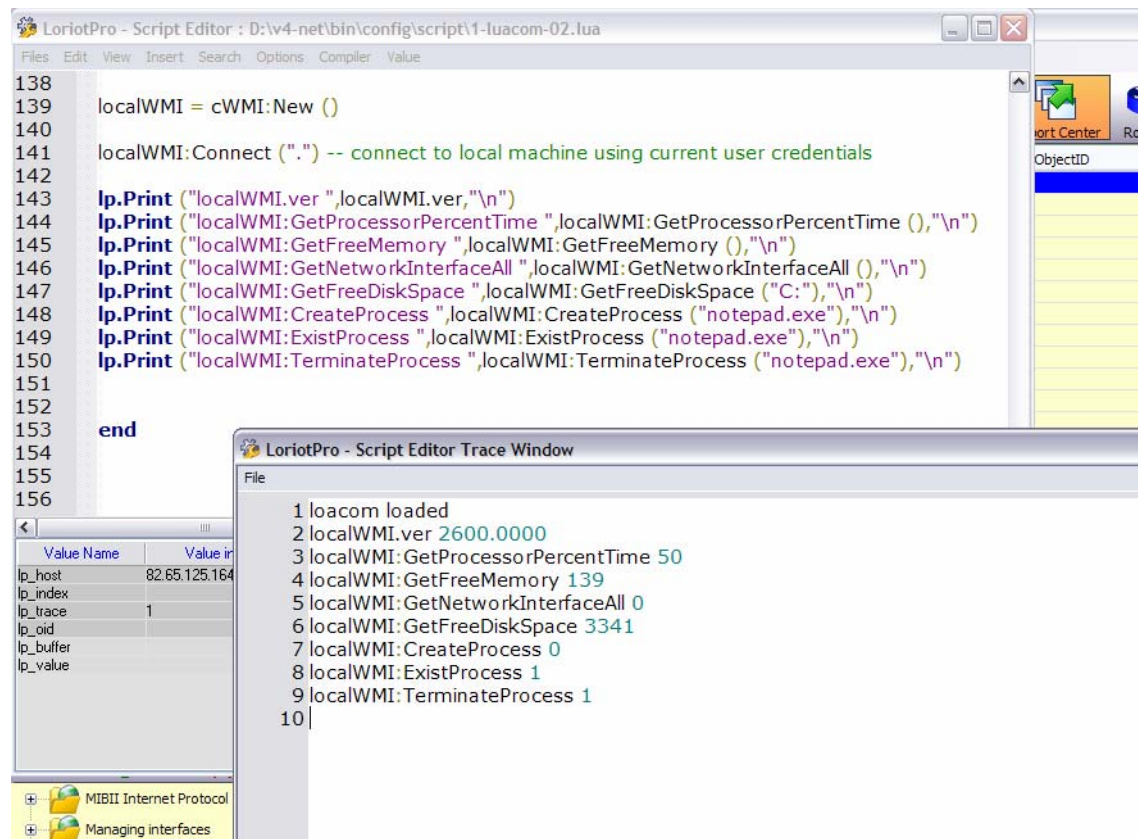
```

```

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

```

Exemple LUA_SOCKET

Cette library (<http://www.cs.princeton.edu/~diego/professional/luasocket/>) est un petit peut complexe mais elle peut être utile. L'exemple suivant envoie un paquet syslog sur le port 516 à LoriotPro.

Exemple :

```
-----  
-- main programme  
-----  
dofile(lp.GetPath().."/config/script/compat-5.1.lua");  
module("socket");  
  
--local socket = require("socket.core");  
  
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", 516  
-- 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("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");
```


LoriotPro V4.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

The screenshot displays the LoriotPro V4.00 Extended Edition interface. The main window is the Script Editor, showing a Lua script for sending a syslog packet. The script includes comments and code for setting host and port, creating a UDP object, sending a packet, and printing the results.

```
29
30
31 -- change here to the host an port you want to contact
32 local host, port = "localhost", 516
33 -- load namespace
34 -- convert host name to ip address
35 local ip = assert(socket.dns.toip(host))
36 -- create a new UDP object
37 local udp = assert(socket.udp())
38 -- contact daytime host
39 assert(udp:sendto("anything from LUA_socket", ip, port))
40 -- retrieve the answer and print results
41 --lp.Print(assert(udp:receive()))
42 lp.Print("send syslog packet \n");
43
44 lp.Print(gcinfo(), " End of script\n");
```

The Trace Window shows the execution steps:

```
File
1 Load library (1537)
2 send syslog packet
3 54 End of script
4
```

The Value Table shows the following data:

Value Name	Value in	Value out
ip_host	127.0.0.1	127.0.0.1
ip_index		
ip_trace	1	1
ip_obj		
ip_buffer	nil	nil
ip_value		

An Information dialog box is displayed with the message "Parsing Ended" and an OK button.

The Events table shows the following data:

TimeStamp	Agent	Ref N°	IP Ref	Alert
Fri Dec 30 17:55:30 2005	Local			anything from LUA
Fri Dec 30 17:52:13 2005	Local			anything

The Mini Map shows a network diagram with several nodes represented by colored shapes (red circles, a red square, a green square, and a pink circle) connected by lines.

The interface also includes a Reports Center, a MIBs Tree, and a status bar at the bottom showing system information and the time 17:55.

Projet Lua_Wizard_LIB

Le package Extended edition contient un exemple et un projet Visual Studio .net 2003 pour vous aider dans la création de vos propre 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}
};

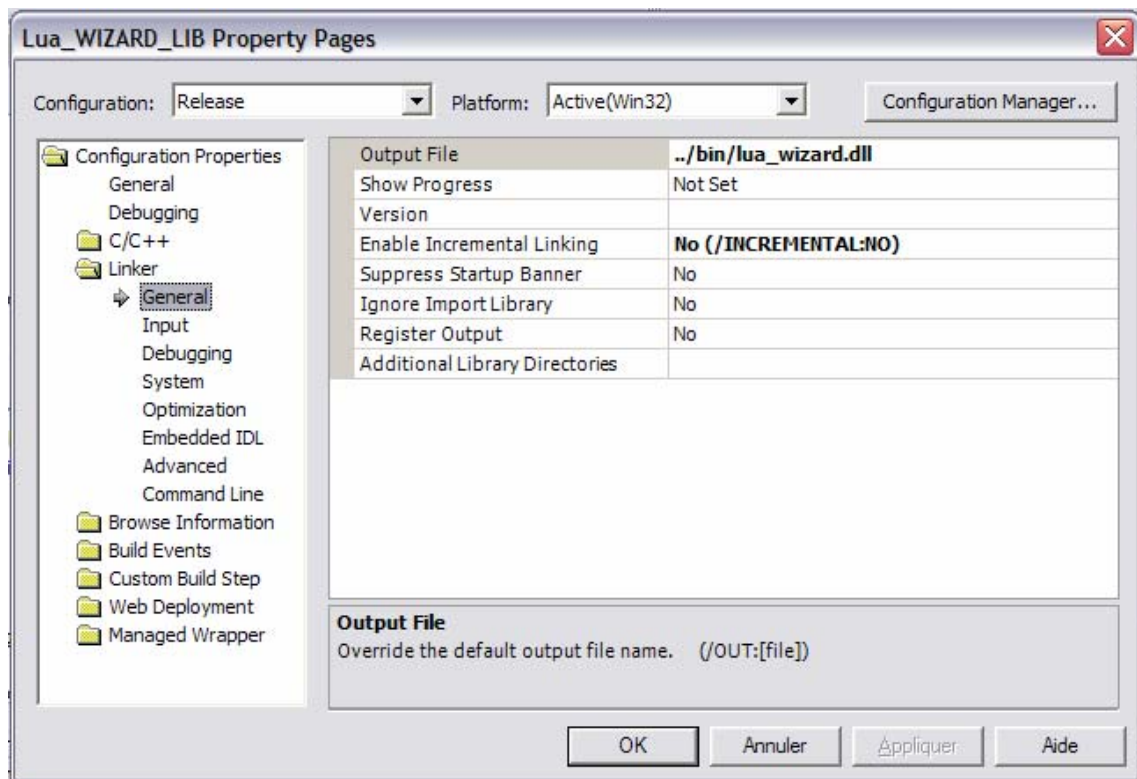
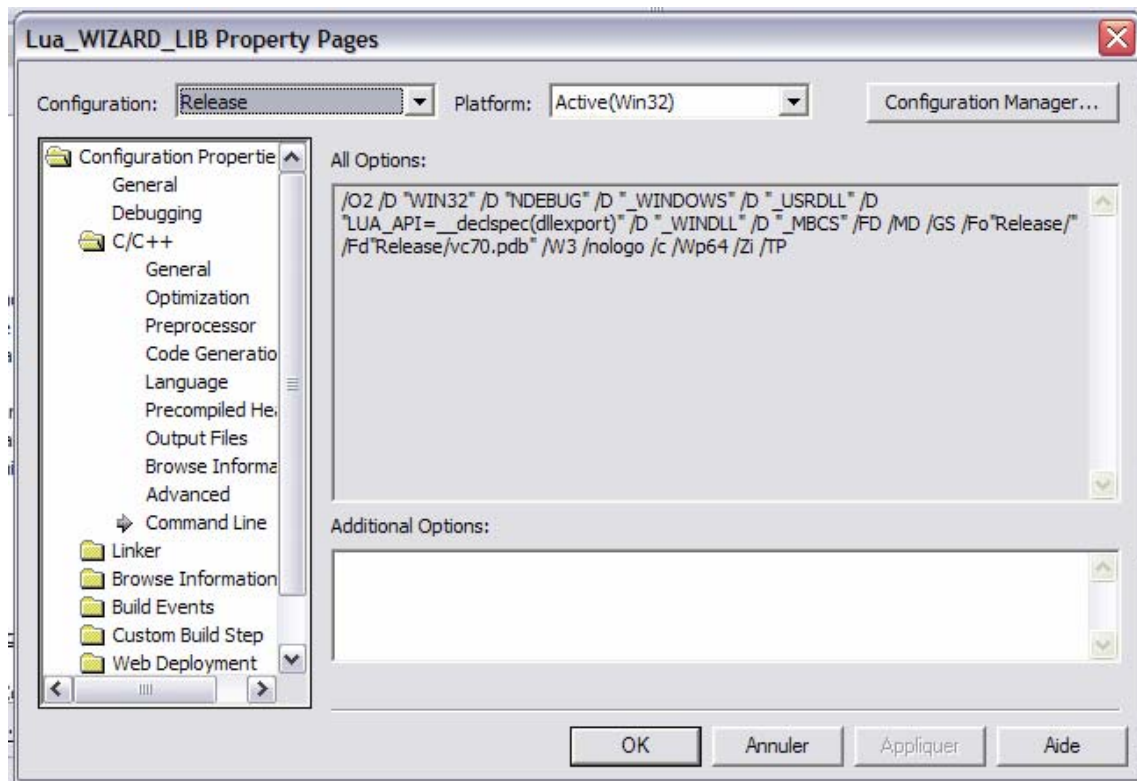
int libinit(lua_State* L)
{
    /* export functions (and leave namespace table on top of stack) */
    //luaL_openlib(L, "lpwiz", func, 0);
    //lua_register(L, "MsgBox", lua_MsgBox);
    luaL_openlib(L, LUA_WIZARD_LIBNAME, wizard_func, 0);
    return 0;
}
```

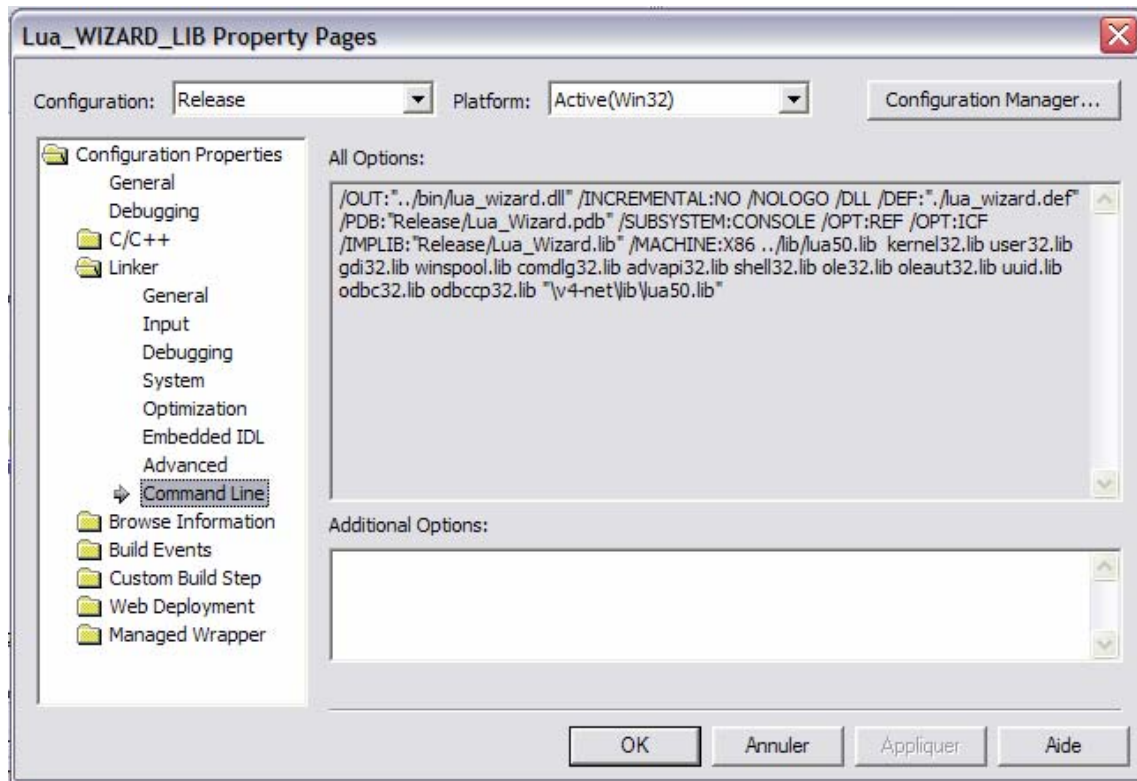
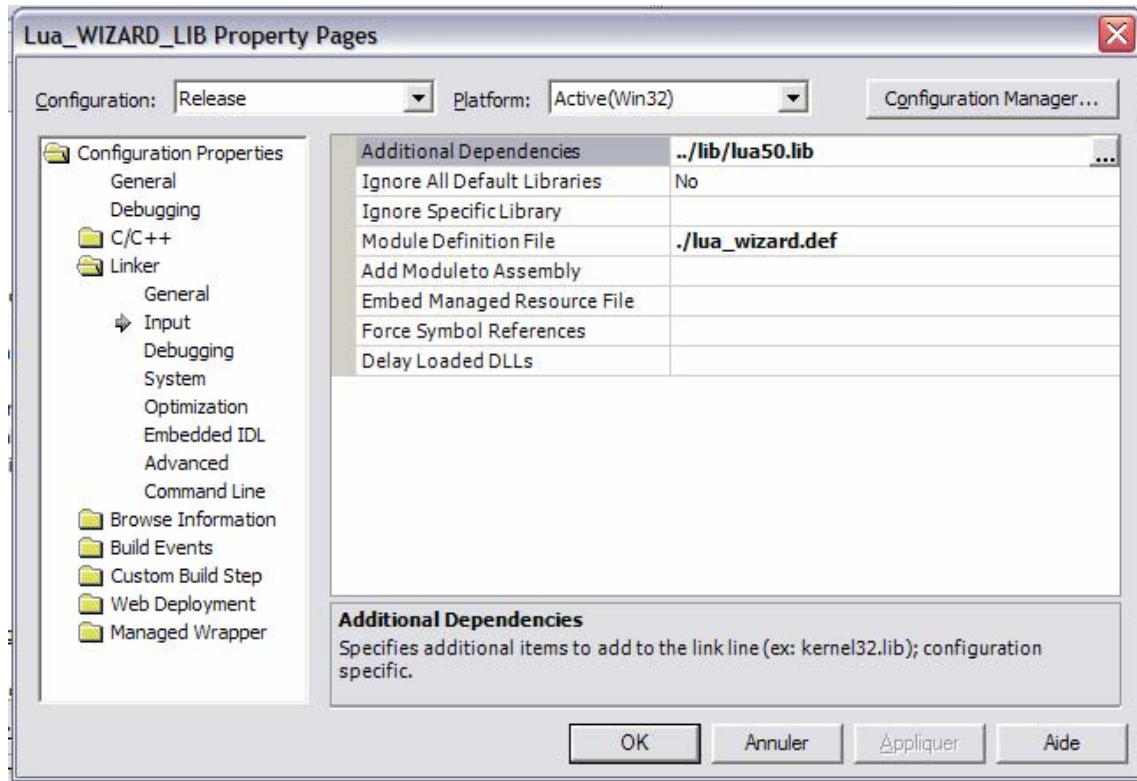
lua_wizard.def

```
LIBRARY    "Lua_WIZARD_LIB"
DESCRIPTION 'LoriotPro LUA 5.0.2 Plug-in MFC Windows Dynamic Link Library'

EXPORTS
    libinit
```

Principales options de compilations





Matrice de compatibilité des libraries LUA

Certaines fonctions se prêtent mal 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					
lp Library noyaux		Oui fonctions nom souhaitables lp.LoadLibrary lp.FreeLibrary	Oui fonctions nom souhaitables lp.LoadLibrary lp.FreeLibrary	Oui	Oui
lpav Library spécialisé pour les objets graphiques des ActiveView		Non	Oui	Oui Un objet graphique doit être sélectionné.	Non
Lpw (lua_lp_wizard.dll) Library spécialisé pour lancer des wizards de sélections.		Non	Non	Oui	Oui
Library externe Lua_odbc Lua_socket Lua_idap Etc...		Non	Non	Oui Si supporté	Oui Si supporté
Interface COM		Non	Non	Oui	Oui

ANNEXES



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.

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.

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

MCours.com