



inWay SA  
102, avenue des Champs-Élysées 75008 PARIS  
☎ +33 1 42 56 41 57 • FAX +33 1 42 89 23 77  
<http://www.inway.fr> • [info@inway.fr](mailto:info@inway.fr)



## *L'expérience*

Une participation à OpenMaster dès 1992 • Une équipe de spécialistes dans tous les domaines liés à l'administration des réseaux et des systèmes • Des experts en développements orientés objets avec une véritable expérience sur le terrain • Une pratique approfondie et un savoir-faire unique sur OpenMaster • Une écoute attentive des besoins des utilisateurs • Une certaine ouverture d'esprit • Une offre de produits en réponse aux attentes du monde réel.

## *L'innovation*

Des produits nouveaux répondant à des contraintes réelles • Une conception fortement structurée • Une architecture orientée objets basée sur des modèles vue/contrôleur • Une approche modulaire et distribuée • Une recherche constante de la simplicité et de l'efficacité • Une économie de moyens, une cohésion, une ergonomie et une qualité de finition inégalées.

## *L'intégration*

Des applications et des services écrits avec le langage et les bibliothèques standard d'OpenMaster • Une compatibilité garantie avec l'existant • Une communication facile avec toute autre application sur OpenMaster.

## *Le partenariat*

Un dialogue permanent avec les utilisateurs • Des rapports clairs avec les intégrateurs • Une amélioration constante des produits • Un soutien sincère de nos partenaires.



## *L'économie*

Une mise en œuvre immédiate • Des performances inégalées • Des besoins en ressources réduits • Un même produit pour de multiples usages • Une seule licence pour tout un projet • Un plan de réduction des coûts par volume.



---

## L'offre

La **SmartSuite** d'*inWay* pour OpenMaster est un ensemble prêt à l'emploi d'applications et d'outils.

### Applications

|               |  |                  |   |
|---------------|--|------------------|---|
| <i>mapper</i> | présentation graphique dynamique du réseau         | <i>explorer</i>  | surveillance réseau/base de données/processus |
| <i>logger</i> | journalisation/présentation/traitement des alarmes | <i>simulator</i> | simulation/contrôle d'un réseau               |
| <i>broker</i> | communications internes entre applications         | <i>mime</i>      | gestionnaire d'objets générique               |

### Sur le Web

|              |                            |
|--------------|----------------------------|
| <i>webit</i> | passerelle PHP avec le Web |
|--------------|----------------------------|

### Outils MIB

|                |                                 |                 |                        |
|----------------|---------------------------------|-----------------|------------------------|
| <i>browser</i> | navigation dans le réseau       | <i>viewer</i>   | consultation du réseau |
| <i>maker</i>   | gestion des instances du réseau | <i>notifier</i> | envoi d'événements     |

### Analyse

|                |  |
|----------------|--|
| <i>cstudio</i> | recherche et définitions des corrélations entre événements avec contraintes de temps |
|----------------|--|

## Les références

La **SmartSuite** d'*inWay* est en opération au coeur de nombreuses solutions d'administration de réseaux.

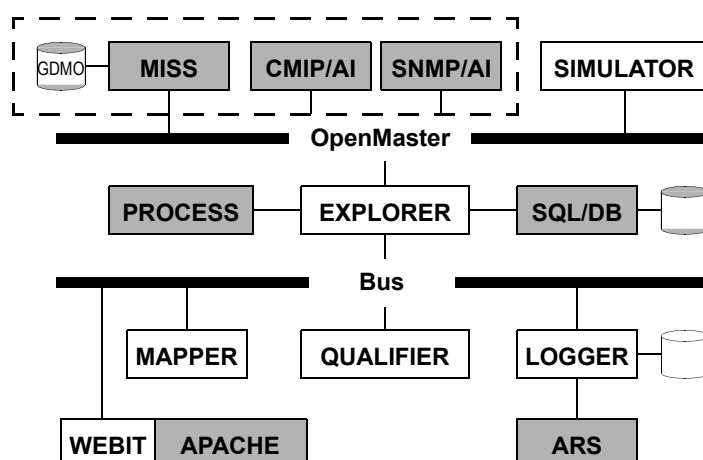


## L'opportunité

Tous les produits d'*inWay* sont disponibles en prêt gratuit. Contactez **Eric Company** au +33 (0)3 29 42 03 59 ou écrivez à [eric.companie@inway.fr](mailto:eric.companie@inway.fr).



# INTEGRATION



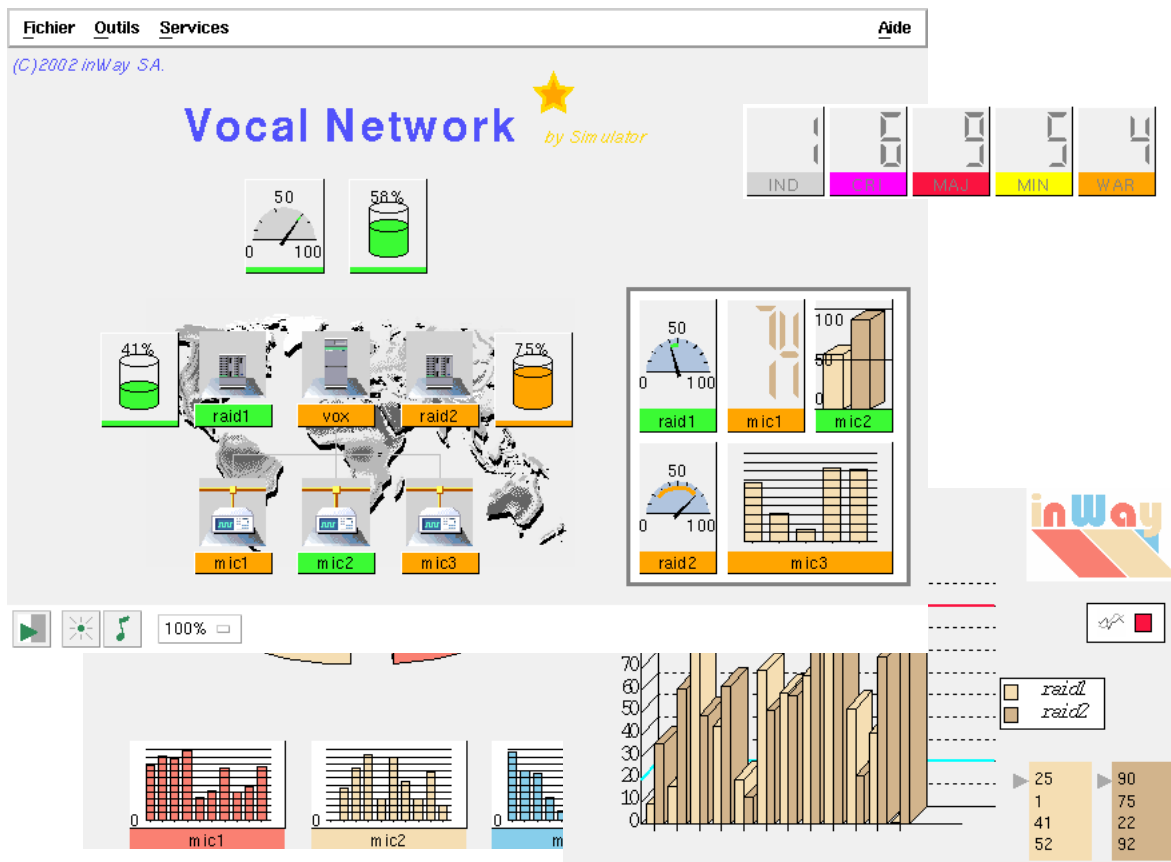
- Les agents intégrateurs SNMP et CMIP sont des composants d'OpenMaster qui traduisent les opérations vers les éléments du réseau et convertissent les notifications d'événements sous une forme normalisée.
- MISS est un service standard offert par OpenMaster qui compile les documents GDMO et RFC. MISS peut encoder et décoder les données définies en ASN.1/BER et distribuer leurs définitions à d'autres applications. L'explorer s'appuie sur ce service pour encoder et décoder les informations échangées avec les éléments du réseau. Les applications graphiques comme les éditeurs de l'explorer et du simulator demandent à MISS la description des syntaxes des données afin de générer automatiquement les objets d'interface.
- L'explorer filtre les événements et surveille la MIB pour le compte d'autres applications comme le mapper ou le logger. Il peut aussi interroger une base de données ou contrôler un autre processus. L'explorer est enfin un véritable système de corrélation capable d'analyser et de relier toutes les informations collectées.
- L'association des agents intégrateurs et de MISS avec l'explorer assure que virtuellement tout composant d'un réseau avec une interface SNMP ou CMIP peut être complètement administré par la SmartSuite. Si un équipement ou une application n'a pas d'interface normalisée, l'explorer est capable de lancer un processus quelconque et de filtrer son flot de sortie. Cette fonction peut servir à surveiller des éléments dotés d'une interface Corba ou ASCII.
- Le mapper permet de représenter un réseau de façon graphique. Il peut afficher des figures simples comme des polygones, des cercles ou des icônes, mais aussi des objets composés comme des histogrammes et des compteurs. Il peut tracer des liens entre toutes les figures. Les objets graphiques sont animés en changeant leurs propriétés en réponse à des messages envoyés par l'explorer ou toute autre application.
- Le qualifier collecte les événements filtrés par l'explorer et génère des alarmes corrélées qui sont journalisées par le logger. Son rôle est d'enrichir le flot des événements tout en le réduisant. Le qualifier est directement inclus dans l'explorer ou le logger.
- Le logger conserve les alarmes et les événements. Son interface graphique montre les alarmes, les événements corrélés, les alarmes historiques dans des tableaux avec des commandes pour les traiter. Il comporte une interface ouverte vers un système de gestion de tickets d'anomalie. Le logger peut qualifier les événements de lui-même ou traiter les alarmes envoyées par le qualifier ou un autre moteur de corrélation plus sophistiqué.
- Le simulator peut émuler tout type de réseau et agir comme un processeur de commande sur le réseau. C'est un composant essentiel pour valider et démontrer des solutions, entraîner les opérateurs dans des conditions extrêmes ou organiser et contrôler les actions sur le réseau.



- Le *webit* donne l'accès complet à l'information élaborée sur la plate-forme d'administration aux applications du Web. Les technologies du Web comme les serveurs Apache et IIS, PHP, les formats HTML et XML, les translateurs XSL et XPATH, .NET, Java, JavaScript et Flash peuvent être librement utilisées pour exploiter tous les services de la SmartSuite.
- Toutes les applications communiquent d'une manière transparente via le bus SML à l'aide du *broker*. Le bus SML est aussi un autre composant standard d'OpenMaster. SML est le langage natif d'OpenMaster employé pour écrire toutes les applications de la SmartSuite.
- La SmartSuite est étonnamment flexible. Tous ses composants sont conçus pour être aussi ouverts et paramétrables que possible. Son but est de fournir une solution toute prête aux intégrateurs, sans développements additionnels.



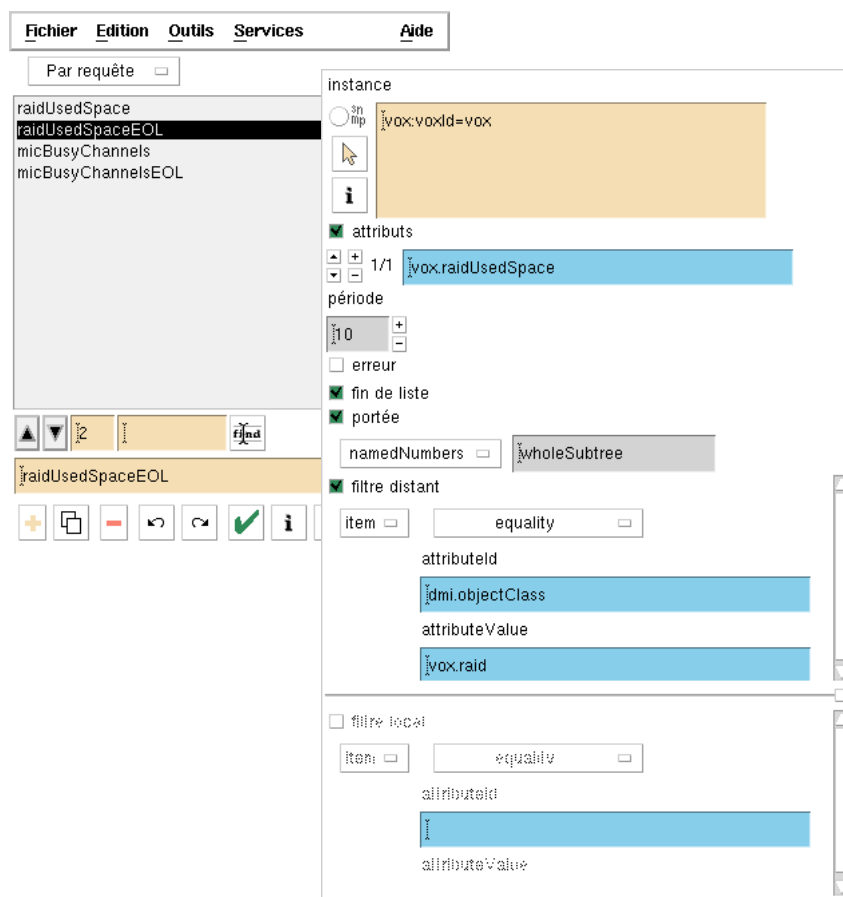
# MAPPER



- présente le réseau sous la forme de cartes graphiques animées constituées de figurines et de liens,
- destiné à être le composant central d'une administration de réseau basée sur OpenMaster,
- résultat d'une longue pratique de la surveillance et la gestion graphique des réseaux,
- construit selon une architecture orientée objets et une boucle de contrôle pilotée par messages,
- peut afficher plusieurs cartes avec ou sans barres de défilement, zoom, menu ou boutons de contrôles,
- les cartes sont dessinées en faisant glisser des objets graphiques à partir de palettes,
- les propriétés graphiques des objets sont éditées à l'aide d'inspecteurs dédiés,
- les figurines et les liens peuvent être animés par tout type d'événement et toute instance de MIB ou de DB,
- les messages d'animation changent les propriétés graphiques des objets sans limitation sur les effets,
- ils peuvent aussi montrer, cacher ou faire clignoter des objets,
- ils peuvent encore déclencher des alertes sonores personnalisées,
- des actions internes ou externes peuvent être lancées à partir de menus contextuels,
- les fichiers des cartes peuvent être générés automatiquement par programme,
- les cartes peuvent être insérées dans des applications spécifiques,
- facile à interfacier avec toute application existante telle que le *logger*,
- livré avec un didacticiel complet et un programme de simulation.



# EXPLORER



- surveille une MIB, une base de données ou un processus distant pour le compte d'autres applications,
- gère une collection de filtres sur des événements réseaux et des requêtes sur la MIB ou une base de données,
- attend des notifications M-EVENT,
- exécute périodiquement des requêtes M-GET ou SQL et des appels à des fonctions distantes,
- envoie des messages à d'autres applications via la *broker* dès qu'une instance ou un événement est reconnu,
- les messages peuvent être accompagnés de données extraites d'un événement ou d'une instance,
- peut regrouper l'information en combinant les messages et corrélérer des événements et des requêtes,
- limite le nombre de requêtes vers la MIB ou une base de données ou un processus distant au strict minimum,
- tout particulièrement conçu et programmé pour passer les filtres le plus vite possible,
- décharge les applications de la tâche complexe de collecte et d'interprétation des informations de gestion du réseau,
- réduit considérablement le trafic et les temps de calcul pour la plate-forme et les applications clientes,
- peut interfacer OpenMaster avec des applications externes à travers le *webit*.



# LOGGER

| #      | Id  | Sév | Rép | Classe | Instance  | F:Date | Heure    |
|--------|-----|-----|-----|--------|-----------|--------|----------|
| 443473 | min | =   | 6   | raid   | vox/raid2 | 03/15  | 17:16:32 |
| 91717  | CRI | =   | 2   | mic    | vox/mic1  | 03/15  | 17:14:10 |
| 908363 | CRI | =   | 4   | raid   | vox/raid1 | 03/15  | 17:16:26 |
| 85717  | war | =   | 2   | raid   | vox/raid1 | 03/15  | 17:14:10 |
| 73504  | CRI | =   | 2   | raid   | vox/raid4 | 03/15  | 17:14:10 |
| 844634 | min | =   | 1   | raid   | vox/raid1 | 03/15  | 17:14:09 |
| 231527 | MAJ | =   | 2   | vox    | vox       | 03/15  | 17:16:36 |
| 465856 | min | =   | 2   | raid   | vox/raid2 | 03/15  | 17:16:31 |
| 434614 | MAJ | =   | 2   | raid   | vox/raid1 | 03/15  | 17:16:38 |
| 690434 | war | =   | 2   | vox    | vox       | 03/15  | 17:16:23 |
| 662036 | MAJ | =   | 2   | mic    | vox/mic2  | 03/15  | 17:16:29 |

| E:Date | Heure    | Id     | Sév | Classe | Instance  | Type | Cause probable |
|--------|----------|--------|-----|--------|-----------|------|----------------|
| 03/15  | 17:13:55 | 276533 | MAJ | raid   | vox/raid2 | Eqp  | receiverFai    |
| 03/15  | 17:14:10 | 968677 | MAJ | raid   | vox/raid2 | Eqp  | receiverFai    |
| 03/15  | 17:16:28 | 681064 | MAJ | raid   | vox/raid2 | Eqp  | receiverFai    |
| 03/15  | 17:16:33 | 74267  | clr | raid   | vox/raid2 | Eqp  | receiverFai    |

- affiche les enregistrements d'alarmes dans des tableaux pouvant contenir du texte coloré ou des images,
- dialogue avec un démon qui corrèle et journalise les alarmes et les événements pour différentes applications,
- contrôle l'accès au journal avec une clé transmise au démon qui détermine ce qui est consultable,
- contrôle les actions sur le journal à l'aide de règles configurables,
- réagit immédiatement à une alarme, n'interroge pas inlassablement le journal,
- signale tout changement par une icône et une alerte sonore personnalisée,
- peut montrer tous les événements corrélés à une alarme donnée,
- peut faire des recherches dans le journal historique des alarmes passées,
- peut afficher le détail d'une alarme, d'une alarme passée ou d'un événement,
- peut ouvrir et clore des tickets d'anomalie,
- peut conserver à l'affichage les alarmes terminées pour un temps donné,
- peut trier une vue par colonne en ordre ascendant ou descendant en un clic,
- peut filtrer une vue sur un champ ou une combinaison logique de champs en un clic,
- peut rechercher une alarme ou un événement à l'aide d'une expression régulière,
- peut compter les alarmes en appliquant des critères programmables,
- peut imprimer les alarmes, les enregistrer dans un fichier pour consultation ultérieure,
- peut déplacer, retailler, cacher ou montrer les colonnes interactivement,
- peut être configuré de multiples façons selon différents profils utilisateurs,
- peut afficher des colonnes dont le contenu est calculé à partir des champs des alarmes ou des événements,
- peut conserver et afficher tout type d'alarme ou d'événement,
- peut être étendu avec plus de fonctions de formatage, de comptage, de filtrage et de tri,
- peut lancer l'exécution de commandes utilisateurs à partir d'un menu,
- peut communiquer avec d'autres applications via le *broker*.



# SIMULATOR

- peut émuler tout élément du réseau ou agir sur des systèmes réels,
- peut servir comme gestionnaire d'objets ou comme processeur de commandes,
- fournit des actions réseaux pour créer, détruire ou modifier des instances (M-CREATE, M-DELETE, M-SET),
- peut extraire des instances (M-GET) et passer leurs attributs à d'autres actions,
- peut aussi envoyer toute action (M-ACTION), tout événement (M-EVENT) ou toute trappe SNMP,
- combine les actions dans des scènes qui exécutent les actions l'une après l'autre,
- groupe les scènes en scénarios qui sont sauvegardés individuellement dans des fichiers textes,
- comprend des actions de contrôle qui, dans une scène, peuvent démarrer ou arrêter une autre scène, ajouter un délai entre deux actions, rompre le flot séquentiel normal avec des instructions IF et GOTO, boucler, passer des valeurs par contexte à d'autres actions, exécuter un code SML et communiquer avec une autre application via le *broker*,
- comprend trois applications : le démon, l'exécutant et l'éditeur,
- le démon déroule les simulations en arrière-plan,
- il peut exécuter plusieurs scénarios et plusieurs scènes dans le même scénario en parallèle,
- l'exécutant dialogue avec le démon,
- il affiche les scénarios chargés en mémoire, leurs descriptions et leurs états de marche et de trace,
- il peut charger et décharger les scénarios, les démarrer et les arrêter, lancer des scènes une à une,
- l'éditeur sert à construire les scénarios, les scènes et les actions,
- les scènes d'un scénario sont présentées sous la forme d'un storyboard,
- les opérations sur la MIB, les actions et les notifications ou les trappes sont décrites à l'aide d'inspecteurs dédiés qui sont générés automatiquement à partir des définitions GDMO et RFC.





# WEBIT

WebIt - Mozilla Firefox

http://www.inway.fr/pub/admin/smallreport.php

Google

inWay WebIt

## Rapport à 15:38:41

L'occupation moyenne de l'espace disque du réseau est de  41%

Niveaux des alarmes actives :

- donne l'accès complet en PHP via le broker aux services et aux applications de la SmartSuite,
- permet d'assurer dynamiquement une mise à disposition transparente sur le Web, via un serveur Apache ou ISS, en toute sécurité et sans déploiement, de toute l'information élaborée sur la plate-forme d'administration du réseau,
- ajoute une tout autre dimension à la SmartSuite avec la possibilité d'utiliser librement toutes les technologies du web comme les serveurs Apache et IIS, PHP, les formats HTML et XML, les translateurs XSL et XPATH, .NET, Java, JavaScript et Flash,
- permet à des développeurs web d'exploiter toute la puissance des services de la SmartSuite et d'exprimer leurs compétences en produisant rapidement une infinité d'applications allant de la simple génération de documents HTML à des plugins Java ou Flash analysant un flot XML.

http://www.inway.fr - :: Flash Log Viewer by inWay :: - Mozilla Firefox

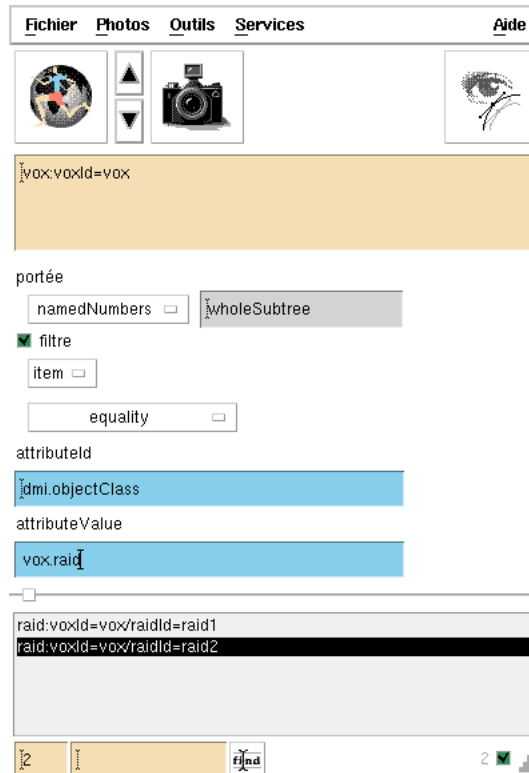
Alarm Log Viewer  
by inWay powered by FLASH

Total : 10 Refresh Preferences

| Id        | Sev | Rep  | Class | Instance  | Type | L:Date Time    | Probable cause    | A:Date Time    | T |
|-----------|-----|------|-------|-----------|------|----------------|-------------------|----------------|---|
| 225704491 | CRI | = 9  | raid  | vox/raid1 | Pro  | 05/22 15:37:16 | corruptData       |                | A |
| 293178504 | war | 1    | vox   | vox       | Com  | 05/22 15:31:16 | 215               |                | A |
| 260514114 | CRI | = 8  | raid  | vox/raid4 | Pro  | 05/22 15:30:07 | outOfMemory       |                | A |
| 190180819 | min | = 4  | mic   | vox/mic2  | Com  | 05/22 15:42:02 | degradedSignal    |                | A |
| 35970734  | min | = 5  | raid  | vox/raid3 | Env  | 05/22 15:35:14 | enclosureDoorOpen |                | A |
| 125612133 | MAJ | = 8  | vox   | vox       | Eqp  | 05/22 15:31:24 | receiverFailure   |                | A |
| 192773107 | min | = 4  | mic   | vox/mic2  | Com  | 05/22 15:41:26 | degradedSignal    |                | A |
| 569405908 | MAJ | = 12 | mic   | vox/mic2  | Eqp  | 05/22 15:36:34 | processorProblem  |                | A |
| 306957309 | MAJ | = 7  | vox   | vox       | Eqp  | 05/22 15:37:47 | receiverFailure   |                | A |
| 681865335 | MAJ | = 8  | mic   | vox/mic2  | Eqp  | 05/22 15:45:43 | processorProblem  | 05/22 15:45:43 | A |



# BROWSER



- navigue dans la MIB,
- admet tout type de portée et de filtre,
- les valeurs d'attribut du filtre sont saisies à l'aide d'inspecteurs dédiés générés dynamiquement d'après le GDMO,
- peut rechercher des instances particulières dans une longue liste avec une expression régulière,
- peut lancer plusieurs requêtes en même temps dans des fenêtres séparées,
- peut sauver les requêtes sur disque,
- tout spécialement conçu pour coopérer avec le *viewer*,
- en un click, le *viewer* peut importer un nom d'instance du *browser*,
- en un click, le *browser* peut demander au *viewer* d'afficher une instance sélectionnée,
- comme service via le *broker*, il peut permettre à toute application de rechercher des noms d'instances dans la MIB.



# VIEWER



- affiche une vue détaillée d'une instance de la MIB,
- peut montrer toute instance,
- les valeurs d'attributs sont mises en forme et affichées à l'aide d'inspecteurs dédiés générés dynamiquement d'après le GDMO,
- peut afficher plusieurs instances en même temps dans des fenêtres séparées,
- peut mettre à jour l'affichage automatiquement,
- peut sauvegarder une instance sur disque,
- tout spécialement conçu pour coopérer avec le *browser*,
- en un click, le *browser* peut demander au *viewer* d'afficher une instance sélectionnée,
- en un click, le *viewer* peut importer un nom d'instance du *browser*,
- comme service via le *broker*, il peut permettre à toute autre application d'afficher des instances de la MIB.



# MAKER

Fichier Outils Services Aide

vox:voxId=vox1

SET

dmi.nameBinding  vox.voxId  
 dmi.objectClass  vox.voxStatus

dmi.allomorphs  vox.voxMicBusyChannels  
 dmi.packages  vox.voxRaidUsedSpace  
 vox.voxLabel  vox.voxTime

vox.voxStatus  
1/1  
operational

- crée, détruit et modifie une instance de la MIB,
- peut opérer sur toute instance,
- les valeurs d'attributs sont saisies à l'aide d'inspecteurs dédiés générés dynamiquement d'après le GDMO,
- peut agir sur plusieurs instances en même temps dans des fenêtres séparées,
- peut sauver une opération sur disque,
- tout spécialement conçu pour coopérer avec le *browser* et le *viewer*,
- en un click, le *maker* peut importer un nom d'instance du *browser*,
- en un click, le *maker* peut demander au *viewer* d'afficher l'instance visée,
- comme service via le *broker*, il peut permettre à toute application d'éditer et d'exécuter des opérations sur la MIB.



# NOTIFIER

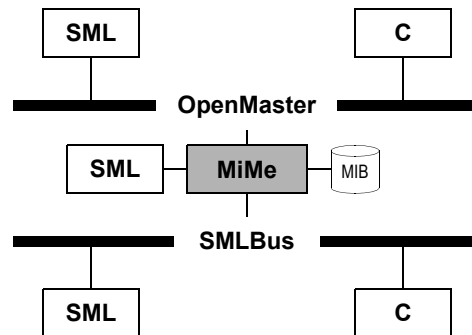
The screenshot shows a web application interface with a menu bar at the top containing 'Fichier', 'Outils', 'Services', and 'Aide'. Below the menu is a header area with a globe icon on the left and a hand icon on the right. A text input field contains 'vox:voxId=vox'. Below this is a blue header for 'dmiequipmentAlarm'. The main content area is a list of configuration options, each with a checkbox and a text input field:

- probableCause
- globalValue
- specificProblems
- perceivedSeverity
- 
- backedUpStatus
- backUpObject
- trendIndication
- 
- thresholdInfo
- notificationIdentifier
- correlatedNotifications
- stateChangeDefinition
- monitoredAttributes
- proposedRepairActions
- additionalText
- additionalInformation

- notifie un événement,
- peut éditer et envoyer tout type d'événement,
- les valeurs des paramètres sont saisies à l'aide d'inspecteurs dédiés générés dynamiquement d'après le GDMO,
- peut éditer plusieurs événements en même temps dans des fenêtres séparées,
- peut sauvegarder une notification sur disque,
- tout spécialement conçu pour coopérer avec le *browser*,
- en un click, le *notifier* peut importer un nom d'instance du *browser*,
- comme service via le *broker*, il peut permettre à toute application d'éditer et d'envoyer des notifications.



## *MIME*



- maintient une MIB en mémoire à l'aide de primitives CMIS,
- lit des milliers d'instances sur disque en quelques secondes,
- se comporte exactement comme un gestionnaire d'objets de la plate-forme,
- prend totalement en charge CMIP,
- peut être facilement étendu afin de bâtir une application spécifique,
- peut servir à écrire des enregistreurs d'alarmes, des agents de proximité, des gestionnaires virtuels, etc.,
- totalement compatible avec CMIS en SML,
- répond aussi à des requêtes directement envoyées en SML via le *broker*,
- permet à des applications de dialoguer en CMIP sans passer par l'infrastructure de communication.



# *BROKER*



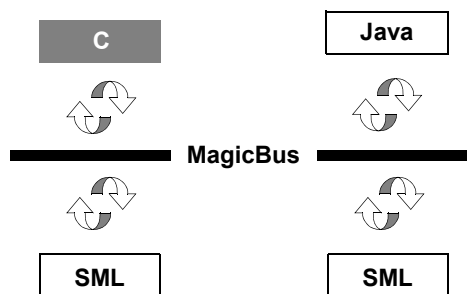
- transforme toute application SML en un service distribué sans code supplémentaire sur le serveur ou le client,
- les connexions, les appels de fonctions distantes, le passage de paramètres et les retours d'erreurs sont transparents,
- toute erreur du côté serveur est capturée et remontée au client comme si elle s'était produite localement,
- interprète un dialogue en SML, en XML-RPC et en PHP sérialisé,
- le *broker* étend radicalement les possibilités d'implémentation en offrant un vrai partage des rôles sans contraintes.

```
;; dans un processus SML, définir une fonction comme d'habitude
? (defun calc:add (x y) (+ x y))
;; l'exporter et s'enregister sur le bus
? (load "smlb") ;charger le broker
? (smlb:export 'calc:add) ;la fonction est maintenant publique
? (smlb:start "Calculator") ;enregistrement sur le bus

;; dans un autre processus SML
? (load "smlb") ;charger le broker
;; importer la fonction et l'exécuter tout simplement
? (smlb:import "Calculator" 'calc:add)
? (calc:add 1 2) ;appel distant avec connexion automatique
= 3
? (calc:add 1) ;une erreur est déclenchée localement
Error: Bad number of arguments: 1
```



# MAGICBUS



- échange des données entre des processus distants par l'envoi de messages,
- les processus peuvent entrer et sortir du *magicbus* à volonté,
- un processus peut s'enregistrer et changer son nom sur le *magicbus* à n'importe quel moment,
- une seule connexion est nécessaire pour communiquer avec plusieurs processus,
- une même interface qui uniformise les échanges de données entre des applications différentes,
- les communications avec le *magicbus* sont asynchrones,
- un processus peut être en même temps un serveur et un client,
- un message peut être envoyé sous forme de notification auquel cas aucune confirmation n'est attendue,
- alternativement, lorsqu'un message est une requête, une réponse est attendue,
- si une réponse n'est pas reçue dans le délai imparti, une erreur est automatiquement remontée,
- un message peut être diffusé vers plusieurs processus enregistrés sous le même nom,
- quand un processus donné disparaît, les messages sont automatiquement redirigés vers un processus de même nom,
- le nombre de processus communicants n'est pas limité,
- la taille des messages échangés par les processus n'est pas limitée,
- une erreur est toujours remontée quand un processus est manquant ou la transmission d'un message est incomplète,
- peut être interfacé par tous les langages de programmation.