

CHAMBREUIL  
Maxime

**Procédure d'installation**  
**du client Big Brother 1.08b**  
**sous W2K**

Juillet / Août 2002

## **I. Installation**

Après avoir téléchargé le client sur <http://www.bb4.com>, lancer le setup.

Welcome

➤ Next

License

➤ Next

User Information

Nom

Entreprise

➤ Next

Destination Location

➤ Parcourir : /Supervision/BB/Client

➤ Next

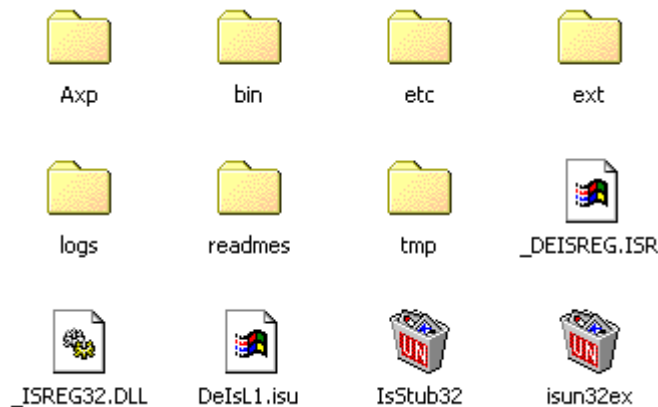
Start Copying Files

➤ Next

Setup Complete

➤ Finish

La fin de l'installation doit lancer le programme de configuration du client automatiquement, si ce n'est pas le cas, allez dans /Supervision/BB/Client/bb18b/bin et lancez « bbntcfg.exe »



« bin » contient l'exécutable du client.

« etc » contient les fichiers de configuration des scripts externes.

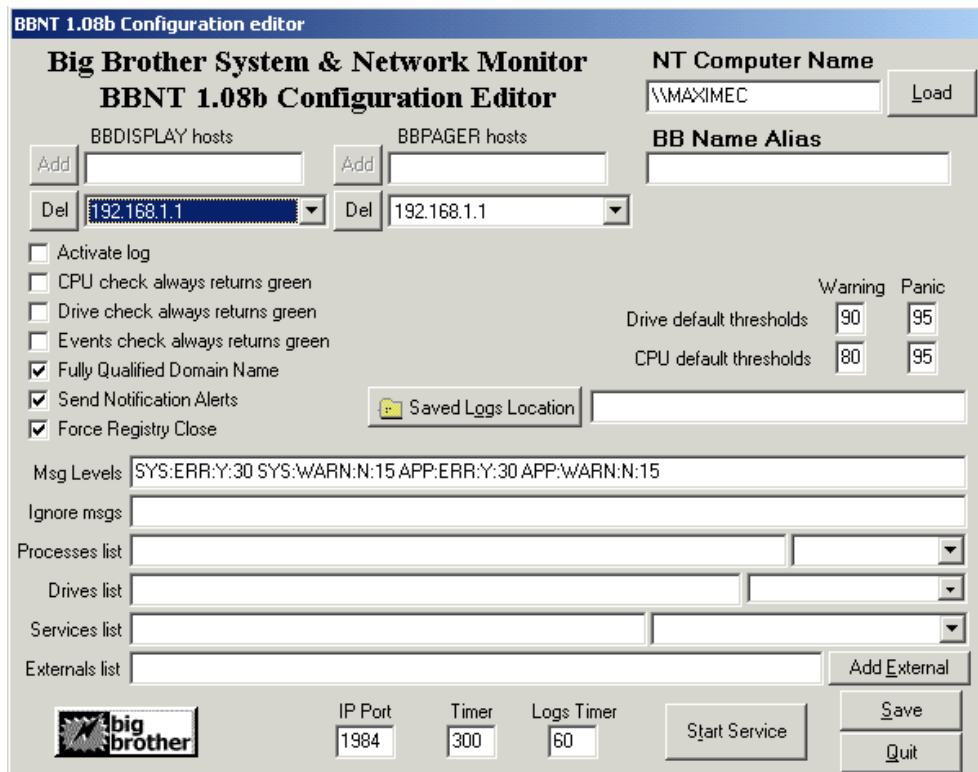
« ext » contient les scripts externes.

« logs » est un répertoire à rajouter, mais ceci est détaillé plus tard.

« tmp » contient les fichiers temporaires.

## II. Configuration

A la fin, la fenêtre de configuration du client doit se lancer automatiquement. Si ce n'est pas le cas, il faut lancer le fichier /Supervision/BB/Client/bb18b/bin/bbntcfg.exe. Vous devriez obtenir cette fenêtre :



Dans la fenêtre, au niveau de BBDISPLAY hosts, vous supprimez l'IP existante ( 192.168.1.1 ) et vous ajoutez celle du serveur.

Vous faites la même chose pour BBPAGER hosts.

Ensuite vous complétez les lignes « Processes list », « Drives list », « Services list », à l'aide des boîtes multichoix situées à chaque fin de ligne, avec les différentes fonctions de la machine que vous souhaitez superviser.

Voici un exemple de configuration :

On enregistre la configuration et on lance le service. Pour le lancement automatique, on va dans Démarrer > Paramètres > Outils d'administration > Services, et on passe le service « Big Brother SNM Client 1.08b » en mode automatique.

Voilà le client est configuré. **Après toute modification, n'oubliez pas de relancer le service Client BB.**

### Ajout de scripts externes

L'ajout de script externe se fait au niveau du client mais c'est le serveur qui affiche les résultats ( cf « Procédure d'installation du serveur BB »). Faites attention à ce que le script soit exécutable sur la plate-forme : Installation de Perl ( cf « Procédure d'installation de MRTG » ).

Un script externe est constitué de 2 fichiers :

- Un fichier Perl, VBScript ou autre langage, contenant le script lui-même
- Un fichier CFG, contenant les paramètres d'utilisation du script.

### **Etape 1 / 5: Configuration du script**

Exemple de fichier CFG pour un script en Perl, qui recherche des chaînes de caractères prédéfinis dans un fichier ( fournie avec le client BB ) :

```
# bb-msgs.cfg
# bb-msgs.pl configuration file
# Copy this file into $BBHOME/ETC
# Vous pouvez définir plusieurs fichiers de logs. Pour chacun, leurs paramètres doivent se
# trouver entre {}

# Parametres :

# logs: Fichier log a vérifier
{
logs: c:\winnt\temp\bb.log

# warnexpr: mots ou expression régulières à rechercher qui déclenche un warning
warnexpr:

# warntime: Durée de warning
warntime: 30

# panicexpr: mots ou expression régulières à rechercher qui déclenche une alerte
panicexpr: ORA-

# panictime: Durée du message d'alerte
panictime: 60

# ignexpr: mots ou expression régulières à ignorer durant le traitement
# Chaque chaîne ou expressions régulières doit être séparée par ';'
ignexpr:

# svcname: Le titre de la colonne à l'affichage sur le serveur
svcname: logs

}
```

### **Etape 2 / 5 : Mise en place des fichiers**

Le script doit être placé dans le répertoire /Supervision/BB/Client/bb18b/ext et son fichier de configuration dans /Supervision/BB/Client/bb18b/cfg.

### Etape 3 / 5 : Création du répertoire logs

Une fois que le script est configuré pour le client, il faut déclarer le script au client BB. Pour cela, il faut créer un répertoire logs au même niveau que etc ; bin ou ext ( dans /Supervision/BB/Client/bb18b ). Au niveau de l'interface de configuration du client, on ajoute une croix pour « Activate logs » et on rentre le repertoire /Supervision/BB/Client/bb18b/logs avec le bouton « Saved Logs Location » :

### Etape 4 / 5 : Déclaration du script au client

Dans la fenêtre de configuration du client, on spécifie le chemin d'accès au fichier Perl grâce au bouton « Add external ». On sauvegarde et on relance le service :

### Etape 5 / 5 : Déclaration du script au serveur

Il nous reste plus qu'à déclarer la nouvelle information, fournie par le script, au serveur ( cf « Procédure d'installation du serveur » ) et à relancer le service client et serveur.

### Ecriture d'un script externe

Une fois que vous savez comment ajouter un script externe, vous voulez sûrement ajouter vos propres scripts.

L'objectif du script est d'écrire dans un fichier l'état de l'évènement que l'on supervise. Ensuite, ce fichier va être lu par le client qui transmettra l'information au serveur.

Le fichier de configuration doit contenir les informations relatives à l'état de l'événement : seuil d'alerte jaune et rouge, le chemin du fichier si votre script est relatif à un fichier, etc...

Un script externe se décompose en 5 parties :

- Une partie de configuration : chemin d'accès au fichier de configuration
- Une partie de traitement du fichier de configuration, ou on teste et on récupère le contenu des différentes variables.
- Le script réel, qui doit instancier les variables \$color et \$msgstatus
- La partie d'écriture dans un fichier temporaire, qui sera lu par le client BB
- Les fonctions de base du script et les votre à rajouter si nécessaire.

Voici le fichier source d'écriture de script, avec les parties à compléter et les commentaires vous concernant :

```
#####
#
#           Configuration
#

# Spécifié le répertoire d'installation du client BB
$BBHOME="d:\\chemin\\vers\\repertoire\\d'installation";

# Nom du fichier de configuration
$cfgfile='bb-test.cfg';

# Chemin du fichier de configuration
$SEP = "\\";
$cfgpath = "$BBHOME${SEP}ETC${SEP}$cfgfile";
$BBTMP = "$BBHOME${SEP}TMP";
$BBSTATLOGS = "$BBHOME${SEP}LOGS";
$NL = "\r\n";

#####
#
#           Lecture du fichier de configuration
#

open (CFG, $cfgpath) || die "ERROR: $! <$cfgpath>$NL";

$first='TRUE';
```



---

```

while (<CFG> ) {
    s/^s+$/g;          # remove trailing whitespaces
    s/^s/ /g;         # change whitespace into space
    s/^s*$/g;
    next if /^s*#/;    # ignore les commentaires
    next if /^s*$/;    # ignore les ligne vides

    if( "$first" eq 'TRUE' ) {
        if( !/}/ ) {
            next;
        }
        $first='FALSE';

        # Paramètres par défaut
        # Rajouter une ligne pour vos paramètres qui nécessitent une valeur par défaut
        $rec = {};
        $rec->{parametre1} = 0;
        next;
    }

    if( "$first" eq 'FALSE' && /}/ ) {
        $first = 'TRUE';
        push @entries, $rec;
        next;
    }

    ($fieldname,$fieldvalue) = split ' ', $_, 2;

    $fieldname =~ tr/A-Z/a-z/;
    $fieldname =~ s/://g;

    # Rajouter une condition pour chacun de vos paramètres
    if( ! ( ($fieldname eq 'parametre1')
            || ($fieldname eq 'parametre2')
          ) ) {
        next;
    }

    $rec->{$fieldname} = $fieldvalue;

    print "$fieldname => $rec->{$fieldname}$NL";
}

# Traitons maintenant chaque fichier log spécifié
unlink "$BBTMP${SEP}.expire";
open EXPIRE , ">>$BBTMP${SEP}.expire" or die "$! - <$file>";

```

```

close EXPIRE;

$AGE_OF_BBTMP__expire = -M "$BBTMP${SEP}.expire";

require "find.pl";

find("$BBTMP");

for $entry ( @entries ) {
    # Test le remplissage des paramètres obligatoires
    # Rajouter un if pour chacun de vos paramètres
    if( $entry->{parametre1} eq " " ) {
        print "$cfgfile misconfigured, missing logs token$NL";
        next;
    }

#####
#
#           Script réel
#       Vous pouvez commencer votre script ici
#       N'oubliez pas d'instancier les variables suivantes :
#   $color: contient la chaine de caractere "green", "yellow" ou "red"
#   $msgstatus: contient le message associé à chaque niveau d'alerte
#

#####
#
#
#       Ecriture du statut et de la couleur dans le fichier temporaire
#

    $BBTMPFILE="$BBSTATLOGS${SEP}$entry->{svcname}.tmp";
    $BBFILE="$BBSTATLOGS${SEP}$entry->{svcname}";
    overwrite($BBTMPFILE,"$color " . localtime(time) . " $msgstatus$NL$NL");
    rename $BBTMPFILE,$BBFILE;
}

#####
#
#       Les fonctions du script
#

```

```
# Fonction utilisée dans l'écriture dans le fichier temporaire
```

```
sub overwrite() {  
my $file = shift(@_);  
my $text = shift(@_);  
  
    open FILE, ">$file" or die "$! - <$file>";  
    print FILE "$text";  
    close FILE;  
}
```

```
# Fonction utilisée dans l'écriture dans le fichier temporaire
```

```
sub wanted {  
    /^MSG.*$/ &&  
    (($dev,$ino,$mode,$nlink,$uid,$gid) = lstat($_)) &&  
    !(-M _ < $AGE_OF_BBTMP__expire) &&  
    unlink($_);  
}
```

```
# Vous pouvez rajouter vos propres fonctions ici
```

```
# Fin de l'exécution du script
```

```
exit 0;
```

Pour les scripts interrogeant la MIB, il est nécessaire d'installer la librairie SNMP de Perl. Vous trouverez différentes versions de cette librairie à cette adresse : <http://www.switch.ch/misc/leinen/snmp/perl/>. Il vous suffit de télécharger le fichier zip, de le décompresser dans <repertoire\_d'install\_de\_perl>/lib. La décompression vous donne un répertoire « SNMP\_session ». Dans ce répertoire, se trouve un répertoire « lib ». Vous copiez tous les fichiers de ce répertoire dans <repertoire\_d'install\_de\_perl>/lib. Voilà la librairie SNMP est installée.

Dans le fichier « bb-editeur.zip » fournit, vous trouverez des éditeurs suivant le type de la réponse SNMP correspondant à l'OID spécifié ( nombre ou chaîne de caractère ). En configurant le fichier .cfg correspondant, on peut ensuite vérifier si le nombre est supérieur ou inférieur à un seuil ou si une chaîne de caractère spéciale est présente dans la réponse. Toutes les précisions quant à la configuration se trouve dans chaque fichier ( bien lire le fichier « README » au préalable ).