

Informations minimums sur le réseau

06/06/2014 17:40 - Icham Sirat

<b>Statut:</b>	Fixed - Corrigé - Implémenté	<b>Début:</b>	11/06/2014
<b>Priorité:</b>	Normale	<b>Echéance:</b>	
<b>Assigné à:</b>	Icham Sirat	<b>% réalisé:</b>	100%
<b>Catégorie:</b>	AbulEduApplicationV1	<b>Temps estimé:</b>	0.00 heure
<b>Version cible:</b>			
<b>Description</b> Avoir une fonction qui calcule : <ul style="list-style-type: none"><li>• l'IP locale de la machine</li><li>• le masque de sous-réseau de l'interface</li><li>• la mac adresse de l'interface associée à l'adresse IP</li></ul>			
<b>Sous-tâches:</b> Task - Tâche # 3725: Test sous linuxFixed - Corrigé - Implémenté Task - Tâche # 3726: Test sur tabletteFixed - Corrigé - Implémenté Task - Tâche # 3727: Test sous windowsFixed - Corrigé - Implémenté			

Historique

#1 - 06/06/2014 17:54 - Icham Sirat

- Statut changé de New - Nouveau à Assigned - En cours
- % réalisé changé de 0 à 60

Voici l'implémentation :

- Une structure pour stockage :

```
struct AbulEduNetworkInfo {
    QHostAddress  HostIp;      /** l'adresse IP */
    QHostAddress  Subnet;      /** le masque de ss-réseau */
    QString       MacAdress;   /** l'adresse MAC associée */
};
```

- Une méthode pour récupérer la structure (avec mise à jour au moment de l'appel)  
Pour pallier le problème de la première adresse MAC trouvée (si multi-config réseau), je retourne l'adresse MAC associée à l'adresse IP trouvée.

```
const AbulEduApplicationV1::AbulEduNetworkInfo &AbulEduApplicationV1::abeGetNetworkInfos()
{
    foreach(const QNetworkInterface &netInterface, QNetworkInterface::allInterfaces())
    {
        foreach (const QNetworkAddressEntry &addrEntry, netInterface.addressEntries()) {
            if(addrEntry.ip().protocol() == QAbstractSocket::IPv4Protocol && addrEntry.ip() != QHostAddress(QHostAddress::LocalHost)) {
                m_networkInfo.HostIp = addrEntry.ip();
                m_networkInfo.Subnet = addrEntry.netmask().toString();
                m_networkInfo.MacAdress = netInterface.hardwareAddress();
            }
        }
    }
    return m_networkInfo;
}
```

- Utilisation

Depuis n'importe où

```
qDebug() << abeApp->abeGetNetworkInfos().HostIp << abeApp->abeGetNetworkInfos().Subnet << abeApp->abeGetNetworkInfos().MacAdress;
```

Bon c'est pas optimal vu qu'à chaque fois que j'appelle `abeApp->abeGetNetworkInfos()`, je reboucle pour remplir toute la structure. mais au moins c'est tout le temps à jour.

## #2 - 06/06/2014 17:54 - Icham Sirat

Ce n'est pas testé donc pas poussé.

## #3 - 11/06/2014 10:11 - Icham Sirat

Les tests sont concluants.

Un peu de nettoyage, debug/log et je pousse :)

## #4 - 11/06/2014 14:53 - Icham Sirat

La structure est publique. De ce fait, on peut en remplir une sans appeler la méthode 3 fois pour avoir toutes les infos.

- Exemple en appelant trois fois la méthode :

```
qDebug() << abeApp->abeGetNetworkInfos().HostIp << abeApp->abeGetNetworkInfos().Subnet << abeApp->abeGetNetworkInfos().MacAdress;
```

- Exemple avec une structure perso (un seul appel à la méthode):

```
AbulEduApplicationV1::AbulEduNetworkInfo networkInfo = abeApp->abeGetNetworkInfos();  
qDebug() << networkInfo.HostIp << networkInfo.Subnet << networkInfo.MacAdress;
```

## #5 - 11/06/2014 14:58 - Icham Sirat

- Statut changé de *Assigned* - En cours à *Fixed* - Corrigé - Implémenté

- Implémenté dans **revno 162**