



StatExplorer

Mode d'emploi administrateur



Table des matières

Maintenance et mise à jour	3
Mises à jour de StatExplorer	3
IPBX Xorcom virtuels, ou appartenant à la gamme CXR/CXE	3
IPBX Xorcom Spark	3
IPBX Xorcom TwinStar	3
Créer des statistiques supplémentaires	6

Maintenance et mise à jour



Attention : les informations présentes dans cette partie sont réservées à un public technique averti. Toute intervention sur la base de données de StatExplorer, ou sur les scripts de maintenance et mise à jour doivent être réalisées par un technicien dument habilité.



Pensez toujours à réaliser une sauvegarde des données de votre IPBX avant toute opération de mise à jour.

Mises à jour de StatExplorer

Les mises à jour de StatExplorer sont réalisées à depuis la console de l'IPBX (accès par SSH) à l'aide des commandes suivantes :

IPBX Xorcom virtuels, ou appartenant à la gamme CXR/CXE

En ssh, saisir la commande

```
| yum update statexplorer
```

IPBX Xorcom Spark

En ssh, saisir la commande

```
| apt update ; apt-get install -only-upgrade statexplorer
```

IPBX Xorcom TwinStar

Les IPBX TwinStar nécessitent une procédure de mise à jour plus complexe. Les étapes de mise à jour sont décrites de façon succincte ci-dessous. Néanmoins, nous vous conseillons de procéder à la mise à jour en association avec le support technique IPConnect.

Sur l'IPBX primaire :

Vérifier que l'état de TwinStar indique bien que l'IPBX est primaire, synchronisé en saisissant la commande :

```
| twinstar-ctl status
```

Le résultat doit être similaire à :

```
AtribanksCount:0
FrozenMaster:    0
InTransition:    false
IsMaster:        true
LocalNodeId:     1
MasterNodeId:    1
NodeCount:       2
PreferredMaster:true
ValidLicense:    true
Version:         1.0
```

La synchronisation DRBD doit également être active et en mode « up to date », vérifiable à l'aide de la commande

```
| drbd-overview
```

dont le résultat doit être identique à :

```
| 0:r0/0  Connected Primary/Secondary UpToDate/UpToDate A r----- /replica
ext3 211G 478M 200G 1%
```

Stopper alors la synchronisation des IPBX et « freezer » les IPBX à l'aide des commandes :

```
| drbdadm disconnect r0
| twinstar-ctl freeze
| twinstar disable-wd
```

Mettre à jour StatExplorer sur l'IPBX primaire à l'aide de la commande :

```
| yum update statexplorer
```

Une fois la mise à jour terminée, vérifier que la synchronisation est bien désactivée en saisissant à nouveau la commande

```
| drbdadm disconnect r0
```

Sur l'IPBX secondaire :

Monter les partitions et services à l'aide des commandes suivantes :

```
| cd /usr/share/twinstar/services.d
| ./10-drbd.run start
| ./15-filesystem.run start
| ./20-mysql.run start
```

```
| ./25-asterisk.run start
| ./30-apache.run start
```

Exécuter maintenant la mise à jour de StatExplorer à l'aide de la commande :

```
| yum update statexplorer
Redémarrez l'IPBX à l'aide de la commande

| reboot
```

Sur les deux IPBX :

Réactiver la synchronisation des IPBX en saisissant les commandes suivantes :

```
| drbdadm connect r0
```

Vérifier alors que la synchronisation est bien lancée à l'aide de la commande :

```
| # cat /proc/drbd
version: 8.4.7-1 (api:1/proto:86-101)
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by
mockbuild@Build64R6, 2016-01-12 13:27:11
 0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent A r-----
    ns:144916 nr:0 dw:10424 dr:167973 al:11 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1
wo:f oos:276976
    [=====>.....] sync'ed: 35.6% (276976/421892)K
    finish: 0:00:11 speed: 24,152 (24,152) K/sec
```

Il est normal que l'IPBX secondaire soit en mode « *inconsistent* ». Dans ce cas, vérifier simplement que la synchronisation est en cours. Une barre de progression doit être affichée avec un pourcentage de synchronisation.

Enfin, sur l'IPBX primaire, exécuter les commandes de réactivation du mode TwinStar à l'aide de :

```
| twinstar-ctl unfreeze
| twinstar enable-wd
```

Créer des statistiques supplémentaires



Attention : les informations présentes dans cette partie sont réservées à un public technique averti. Les modifications des fichiers de StatExplorer peuvent provoquer la perte de données irréversibles.



Pensez toujours à réaliser une sauvegarde des données de votre IPBX avant toute opération sur les fichiers.

StatExplorer a été conçu pour permettre l'ajout de nouveaux modules de statistiques par une équipe de développement directement sur demande du client final.

L'ajout d'une statistique se fait en ajoutant un fichier Javascript dédié dans le repertoire **/usr/share/ombutel/www/statexplorer/js/**

Le fichier doit être nommé selon le format suivant : **plugin_xxxx_nom.js** ou **xxxx** représente la position de la statistique dans le menu déroulant.

Le fichier javascript doit contenir la structure de code suivante :

```
StatExplorer.RegisterPlugin(new (function()
{
    var that = this;
    this.id = "unique_name_without_spaces";
    this.name = "Title of the plugin";
    this.description = "Description of the plugin";

    //This method is in charge of getting data from the StatExplorer API
    //This methode call the callback method as soon as the data are
    available. Callback accep 3 arguments: data, errorcode, errortext
    var getData = function (filter, callback)
    {
        //Todo: Get data from StatExplorer API or other access, using
        filter object.
    };

    //This method is in charge of building the graphics.
    //For statexplorer, we use plotly.js
    var plotGraph = function (elem, data)
    {
        //Todo: build graphics and place the graphics in "elem" dom
        element here using data
    };

    //This method is in charge of building the table with all the data
    inside.
```



```

    var drawTable = function (elem, data)
    {
        //Todo: build table and place the table in "elem" dom element
        here using data
    };

    //This method is in charge of building the whole result, buy getting
    data, then building plot and table.
    this.Display = function(graphelem, tablelem, filter, callback,
    errorCallback)
    {
        //Tell the customer that we are calculating data
        msgbox.ShowMsgBoxInfo(true, display("results.loading"), 0,
        "loading_results", false);
        //Keep references of callback.
        var mcb = callback; var mcberror = errorCallback;
        //Get data from API
        getData(filter, function (data, errorcode, errortext)
        {
            if (data !== null)
            {
                //display graphs and tables
                plotGraph(graphelem, data);
                drawTable(tablelem, data);
                //Call callbacks
                if (mcb !== undefined && mcb !== null)
                    mcb();
                //hide the loading message
                msgbox.HideMsgBoxByName("loading_results");
            }
            else
            {
                //Call error callbacks
                if (mcberror !== undefined && mcberror !== null)
                    mcberror(errorcode, errortext);
                //hide the loading message
                msgbox.HideMsgBoxByName("loading_results");
            }
        });
    };

    //This method is called by StatExplorer to get data in CSV format for
    export.
    this.GetCSV = function ()
    {
        var csv = "";
        //Todo: type export code here
        return csv;
    };
    })());

```

La méthode **getData** doit ainsi récupérer les données à afficher depuis l'API ou depuis une source externe, et appeler le callback en fournissant les arguments suivants :

- Les données chargées (ou null en cas d'erreur)
- Le code d'erreur (ou null)
- La description de l'erreur (ou null)

Les méthodes **plotGraph** et **drawTable** sont en charge de l'affichage des graphiques et tableaux de résultats. Les paramètres de ces méthodes sont identiques :

- Element DOM où afficher le graphique/tableau (exemple : une div)
- Les données à traiter.

La méthode **Display** est appelée par StatExplorer lors de l'affichage de la statistique.

La méthode **GetCSV** est appelée par StatExplorer lors de l'export de résultats au format CSV.

Vous pouvez modifier la structure du fichier, mais les méthodes **Display et **GetCSV** sont obligatoires.**