



API CallBox (CTI)



Table des matières

1	Préambule	3
1.1	Définitions	3
1.2	Description	4
1.2.1	Méthodes disponibles	4
1.2.2	Protocoles SSL supportés	4
2	Variables réseau pour l'utilisation de l'API CallBox	5
2.1	Utilisation hors du réseau local du serveur CallBox	5
2.2	Utilisation dans le réseau local du serveur CallBox	5
3	API CallBox via JSON	6
3.1	Authentification/Confidentialité	6
3.1.1	Authentification via PSK	6
3.2	Méthodes	6
3.2.1	Méthode GET	6
3.2.2	Méthode POST	6
3.3	Acquittement	7
3.4	Fonctions	7
3.4.1	Supervision	7
3.4.2	Historique d'appel	8
3.4.3	Lancement d'appel	9
3.4.4	Transfert d'appel	9
3.4.5	Re-routage	11
3.4.6	Pause	11
3.4.7	Switch	12
3.4.8	Téléchargement d'un enregistrement d'appel	12
3.4.9	Ecoute d'un enregistrement d'appel	12
3.5	Exemple d'utilisation	13
3.5.1	Web Call Back	13
4	API CallBox via XML	14
4.1	Pré-requis	14
4.2	Connexion à l'API CallBox XML	14
4.3	Notification d'appel (remontée de fiche)	14
4.4	Lancement d'appel (click-to-call)	14
4.5	Déconnexion	15
4.6	Exemple d'utilisation	15
5	Liens tel : callto : et callbox :	16
5.1	Mise en oeuvre d'un lien d'appel dans une page HTML	16
6	API CallBox Web	17
6.1	Pré-requis	17
6.2	Notification d'appel (remontée de fiche)	17
6.3	Lancement d'appel (click-to-call)	17
6.3.1	Web application	17
7	API CallBox via exécutable	18
8	Contrôle d'un élément de routage "Switch App"	19



9	Import automatique de contacts	20
9.1	Fichier CSV	20
9.1.1	Structure du fichier	20
9.1.2	Récupération automatique des données	20
9.2	Contacts Gmail™	21
9.3	A la demande	21
10	Contacts partagés	22
10.1	Contacts partagés CardDAV	22
10.1.1	Correspondance Champs CardDAV / Contacts CallBox	22
10.2	Contacts partagés LDAP(S)	22
10.2.1	Configuration réseau	22
10.2.2	Paramètres LDAP(S) CallBox	23
10.2.3	Correspondance Champs LDAP(S) / Contacts CallBox	23



1 Préambule

Ce manuel détaille le fonctionnement de l'API CallBox permettant de mettre en oeuvre des fonctionnalités de type CTI (Couplage Téléphonie Informatique) au sein d'un logiciel tierce partie.

Si le développement réalisé est fonctionnel, votre application sera référencée parmi les logiciels compatibles API CallBox.

Pour toute question technique concernant l'utilisation de l'API CallBox, [contactez Thinkro System](#).

1.1 Définitions

API Une interface de programmation (Application Programming Interface ou API) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine. Du point de vue technique une API est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service.

CTI Le couplage téléphonie informatique (CTI) est l'ensemble des techniques permettant la mise en oeuvre d'applications reposant sur un interfonctionnement d'applicatifs informatiques et d'applicatifs téléphoniques. Il s'inscrit dans la logique de convergence numérique. Ce dispositif reliant un centre d'appel interne ou externe au système informatique d'une entreprise, apporte aux sociétés la possibilité d'utiliser les ressources du système d'information et d'Internet, afin d'améliorer le service rendu aux clients et la productivité d'un centre d'appels. Ceci est rendu possible par l'automatisation de certaines tâches et la possibilité d'instaurer une personnalisation poussée de la relation avec le client via les informations mises à disposition par ce système. Il rend possible l'affichage simultané sur l'écran du télé-opérateur de données (fiche du client et script de l'entretien) ainsi que la présentation de l'appel sur son poste téléphonique. Par exemple, l'identification d'un numéro de téléphone permettra de retrouver les informations concernant

l'appelant pendant que l'appel est transféré au bon interlocuteur. La composition automatique de numéros à partir d'une liste d'adresses constitue une autre application.

JSON JavaScript Object Notation (JSON) est un format de données textuelles qui permet de représenter de l'information structurée comme le permet XML par exemple. Un document JSON rassemble de l'information accompagnée d'étiquettes ou labels permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci. Deux types d'éléments structurels composent un fichier JSON : des ensembles de paires "nom" (alias "clé") / valeur et des listes ordonnées de valeurs.

Socket Dans le contexte des logiciels, on peut le traduire par "connecteur réseau". Apparu dans les systèmes UNIX, un socket est un élément logiciel qui est aujourd'hui répandu dans la plupart des systèmes d'exploitation. Il s'agit d'une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau. Il lui sera ainsi par exemple aisé d'établir une session TCP, puis de recevoir et d'expédier des données grâce à elle. Cela simplifie sa tâche car cette couche logicielle, de laquelle il requiert des services en appelant des fonctions, masque le nécessaire travail de gestion du réseau, pris en charge par le système. Le terme socket désigne en pratique chaque variable employée dans un programme afin de gérer l'une des sessions.



1.2 Description

Liste des fonctionnalités de l'API CallBox de Couplage Téléphonie Informatique :

- Lancement d'appell (click-to-call)
- Notification d'appel (remontée de fiche)
- Pilotage de la Pause des utilisateurs
- Pilotage d'un Switch ou d'un Switch App de routage d'appel
- Supervision détaillée des appels en temps réel
- Journal/historique détaillé des appels
- Accès aux enregistrements des communications téléphoniques
- Import automatique de contacts

1.2.1 Méthodes disponibles

- JSON (recommandé)
- XML
- Web
- Exécutable
- Microsoft® Windows® TAPI

1.2.2 Protocoles SSL supportés

- SSLv2
- SSLv3
- TLSv1



2 Variables réseau pour l'utilisation de l'API CallBox

2.1 Utilisation hors du réseau local du serveur CallBox

Pré-requis : pour utiliser l'API CallBox en dehors du réseau local du serveur CallBox, rendez-vous dans votre espace installateur CallBox et autorisez l'IP WAN.

En mode distant, les variables suivantes, lorsqu'elles sont évoquées dans ce guide, ont pour valeur :

[IP-SERVEUR-CALLBOX] = Adresse IP WAN du serveur CallBox ou nom de domaine pour faciliter les changements d'IP
[PORT-ADMINISTRATION] = Valeur du champ "Port Administration" renseignée dans l'espace installateur (par défaut : 480)

2.2 Utilisation dans le réseau local du serveur CallBox

En mode local, les variables suivantes, lorsqu'elles sont évoquées dans ce guide, ont pour valeur :

[IP-SERVEUR-CALLBOX] = Adresse IP LAN du serveur CallBox
[PORT-ADMINISTRATION] = 80



3 API CallBox via JSON

3.1 Authentification/Confidentialité

La confidentialité des échanges avec l'API est assuré par du filtrage IP.

Il est fortement conseillé de crypter les échanges via HTTPS.

La confidentialité peut encore être renforcée via une authentification de type PSK (*Pre-Shared Key*).

3.1.1 Authentification via PSK

3.1.1.1 Préalablement

- Définissez la valeur de la clé partagée (PSK) dans CallBox UI > perspective Configuration > vue Paramètres > section Configuration API > champ "API CallBox JSON : PSK".

3.1.1.2 Utilisation

- Dans chaque requête JSON, ajoutez la valeur de la clé partagée (PSK) comme dans l'exemple suivant :

```
{  
  "function": "supervision"  
  "PSK": "xxxxxxxxxxxxxxxxxxxxxx"  
}
```

3.2 Méthodes

3.2.1 Méthode GET

Cette méthode fonctionne dans un navigateur Internet.

Construction d'une demande en GET (HTTP ou HTTPS) :

```
https://[IP-SERVEUR-CALLBOX]:[PORT-ADMINISTRATION]/callbox/api_json.php?query={"function":"pause",  
  "user":"*11","set":"enabled"}
```

3.2.2 Méthode POST

Cette méthode est utilisable en ligne de commande (Linux/macOS).

Construction d'une demande en POST (HTTP ou HTTPS) :



```
curl -X POST -d 'query={"function": "pause","user": "*11","set": "enabled"}' https://[IP-SERVEUR-CALLBOX]:[PORT-ADMINISTRATION]/callbox/api_json.php
```

3.3 Acquittement

Chaque demande JSON fait l'objet d'un acquittement.

- Demande de données correctement réalisée :
Retour d'un fichier JSON contenant les données souhaitées.
- Demande de type action correctement réalisée :

```
{  
  "status": "ok"  
}
```

- Demande non-exécutée :

```
{  
  "status": "error"  
}
```

3.4 Fonctions

3.4.1 Supervision

3.4.1.1 Requêtes

- Demande des informations de supervision :

```
{  
  "function": "supervision"  
}
```

- Demande des informations de supervision d'un utilisateur déterminé :

```
{  
  "function": "supervision",  
  "user": "*15"  
}
```




Dans cet exemple, l'API retourne les informations de supervision de l'utilisateur *15 uniquement.

3.4.1.2 Données retournées

Signification des valeurs prises par certaines données retournées par l'API :

Donnée "sens" : Sens d'appel

- Valeur "1" : Appel externe entrant
- Valeur "2" : Appel externe sortant
- Valeur "3" : Appel interne
- Valeur "4" : Appel via Interconnexion sortant
- Valeur "5" : Appel via Interconnexion entrant
- Valeur "6" : Appel Inter-CallBox sortant
- Valeur "7" : Appel Inter-CallBox entrant
- Valeur "8" : Appel lancé par un module Numérotation depuis un module File
- Valeur "9" : Appel du système de conférence téléphonique
- Valeur "10" : Appel du système de visio-conférence
- Valeur "11" : Transfert d'appel

3.4.2 Historique d'appel

3.4.2.1 Requêtes

- Demande de l'historique d'appel d'une journée :

```
{  
  "function": "histo",  
  "day": "2018-01-20"  
}
```

Dans cet exemple, l'API retourne l'historique d'appel de la journée du 20/01/2018.

- Demande de lignes spécifiques de l'historique d'appel :

```
{  
  "function": "histo",  
  "start_id": "1",  
  "limit": "50"  
}
```

Dans cet exemple, l'API retourne 50 lignes d'historique d'appel depuis la première ligne.

Note : limit max. = 5000



3.4.2.2 Données retournées

Signification des valeurs prises par certaines données retournées par l'API :

Donnée "sens" : Sens d'appel

- Valeur "1" : Appel externe entrant
- Valeur "2" : Appel externe sortant
- Valeur "3" : Appel interne
- Valeur "4" : Appel via Interconnexion sortant
- Valeur "5" : Appel via Interconnexion entrant
- Valeur "6" : Appel Inter-CallBox sortant
- Valeur "7" : Appel Inter-CallBox entrant
- Valeur "8" : Appel lancé par un module Numérotation depuis un module File
- Valeur "9" : Appel du système de conférence téléphonique
- Valeur "10" : Appel du système de visio-conférence
- Valeur "11" : Transfert d'appel

3.4.3 Lancement d'appel

- Lancement d'appel depuis un poste utilisateur déterminé :

```
{  
  "function": "call",  
  "user": "*15",  
  "dial": "06XXXXXXXX"  
  [Optionnel] "sipheader": "Alert-Info:answer-after=0"  
}
```

Dans cet exemple, l'API compose le numéro 06XXXXXXXX depuis le poste utilisateur *15. Ici, le champ optionnel indique au poste téléphonique d'effectuer un décroché automatique.

3.4.4 Transfert d'appel

3.4.4.1 Transfert supervisé

- Transfert supervisé d'un appel :

```
{  
  "function": "transfer",  
  "type": "attended",  
  "channel": "SIP/*10-0001e95d",  
  "destination": "*15"  
}
```

L'identifiant du *channel* initiateur du transfert est présent dans le tableau *chanlist* d'un appel de la fonction de Supervision qui apparaît sous la forme :



```
"calls": {
  "1576752643.626915": {
    "sens": "1",
    "num": "06XXXXXXXX",
    "record": "1",
    "name": "John DOE",
    "duration": "54",
    "chanlist": [
      "SIP/trunk_sip_25-0001e95e",
      "SIP/*10-0001e95d"
    ],
    "state": "Up"
  }
},
```

Dans cet exemple, l'API effectue un transfert supervisé d'appel, initié par l'utilisateur *10, vers le poste utilisateur *15 qui se retrouve en relation avec l'utilisateur *10 pendant que l'appelant 06XXXXXXXX est en attente avant la finalisation du transfert.

3.4.4.2 Transfert aveugle

- Transfert aveugle d'un appel :

```
{
  "function": "transfer",
  "type": "blind",
  "channel": "SIP/*10-0001e95d",
  "destination": "*15"
}
```

L'identifiant du *channel* initiateur du transfert est présent dans le tableau *chanlist* d'un appel de la fonction de Supervision qui apparaît sous la forme :

```
"calls": {
  "1576752643.626915": {
    "sens": "1",
    "num": "06XXXXXXXX",
    "record": "1",
    "name": "John DOE",
    "duration": "54",
    "chanlist": [
      "SIP/trunk_sip_25-0001e95e",
      "SIP/*10-0001e95d"
    ],
    "state": "Up"
  }
},
```

Dans cet exemple, l'API effectue un transfert aveugle d'appel, initié par l'utilisateur *10, vers le poste utilisateur *15 qui se retrouve en relation avec l'appelant 06XXXXXXXX.



3.4.5 Re-routage

Attention : cette fonction est très puissante et peut, par exemple, aboutir à interrompre un appel en cours si elle est mal employée.

- Modification du routage d'un appel :

```
{
  "function": "redirect",
  "channel": "SIP/trunk_sip_25-0001e95e",
  "destination": "*15",
  [Optionnel] "sipheader": "Alert-Info:answer-after=0"
}
```

L'identifiant du *channel* est présent dans le tableau *chanlist* d'un appel de la fonction de Supervision qui apparaît sous la forme :

```
"calls": {
  "1576752643.626915": {
    "sens": "1",
    "num": "06XXXXXXXX",
    "record": "1",
    "name": "John DOE",
    "duration": "54",
    "chanlist": [
      "SIP/trunk_sip_25-0001e95e"
    ],
    "state": "Up"
  }
},
```

Dans cet exemple, l'API re-route l'appel du 06XXXXXXXX vers le poste de l'utilisateur *15. Ici, le champ optionnel indique au poste téléphonique d'effectuer un décroché automatique.

3.4.6 Pause

- Mise en pause d'un utilisateur :

```
{
  "function": "pause",
  "user": "*15",
  "set": "enabled"
}
```

Dans cet exemple, l'utilisateur *15 est mis en pause.

- Sortie de pause d'un utilisateur :

```
{
  "function": "pause",
  "user": "*15",
}
```



```
"set": "disabled"
}
```

Dans cet exemple, l'utilisateur *15 est sorti de pause.

3.4.7 Switch

- Modification de la position d'un switch :

```
{
  "function": "switch",
  "number": "10",
  "position": "0"
}
```

Dans cet exemple, le switch numéro 10 sera en position 0 (Position par défaut).

3.4.8 Téléchargement d'un enregistrement d'appel

- Téléchargement d'un enregistrement d'appel :

```
{
  "function": "recordFile",
  "id": "5555"
}
```

Dans cet exemple, l'enregistrement numéro 5555 est téléchargé.

Note : pour obtenir l'ID d'un enregistrement d'appel, utilisez la fonction "histo", puis recherchez la clé "id" correspondante dans le fichier JSON retourné.

3.4.9 Ecoute d'un enregistrement d'appel

- Ecouter un enregistrement d'appel depuis un téléphone :

```
{
  "function": "recordListen",
  "id": "5555",
  "user": "*15"
}
```

Dans cet exemple, l'API lance la lecture de l'enregistrement numéro 5555 sur le téléphone de l'utilisateur *15.

Note : pour obtenir l'ID d'un enregistrement d'appel, utilisez la fonction "histo", puis recherchez la clé "id" correspondante dans le fichier JSON retourné.



3.5 Exemple d'utilisation

3.5.1 Web Call Back

La fonction de Web Call Back permet à un internaute d'être rappelé au numéro qu'il a préalablement saisi sur le site web de l'entreprise.

Cliquez ci-dessous pour télécharger un fichier d'exemple d'intégration de la fonction de rappel automatique :
[Web Call Back - Fichier PHP d'exemple](#)



4 API CallBox via XML

4.1 Pré-requis

Sur le poste informatique de l'utilisateur :

1. Installez et démarrez le logiciel CallBox UI.
2. Rendez-vous dans la perspective "Supervision" du logiciel CallBox UI et identifiez-vous en tant qu'utilisateur CallBox.

4.2 Connexion à l'API CallBox XML

Pour utiliser l'API CallBox XML, connectez-vous sur le socket `localhost:3009` (read/write).

4.3 Notification d'appel (remontée de fiche)

Lors de la réception d'un appel sur le téléphone de l'utilisateur CallBox, l'API CallBox transmet un Event XML sur le socket `localhost:3009` :

```
<event>
  <call sens="IN" action="HANGUP" number="0954933277" name="Mr Martin" id="00032a" />
</event>
```

Un Event XML est toujours terminé par une ligne vide.

Attributs et valeurs possibles contenues dans l'Event XML que vous recevez :

- sens : IN (appel entrant), OUT (appel sortant) ou OTHER (sens d'appel indéterminé).
- action : RING (sonnerie), ESTABLISHED (communication établie) ou HANGUP (appel raccroché).
- number : le numéro de téléphone de l'appelant.
- name : le nom du contact correspondant à ce numéro de téléphone du carnet d'adresses CallBox.
- id : un identifiant unique d'appel.

4.4 Lancement d'appel (click-to-call)

Pour effectuer le lancement d'un appel depuis le poste d'un utilisateur CallBox, envoyez une Action XML sur le socket `localhost:3009` :

```
<action>
  <call number="0662140255" />
</action>
```

Une Action XML est toujours terminée par une ligne vide.



4.5 Déconnexion

Pour se déconnecter de l'API CallBox XML, lancez une Action XML sur le socket `localhost:3009` :

```
<action>
  <quit />
</action>
```

Une Action XML est toujours terminée par une ligne vide.

4.6 Exemple d'utilisation

Sur un système exécutant Microsoft® Windows®, en utilisant l'outil logiciel "AutoHotKey", il est possible de configurer un raccourci clavier permettant de lancer un appel sur la sélection d'un numéro de téléphone réalisée depuis n'importe quelle application.

Téléchargez le code source de l'exemple depuis <http://mir.thinkrosystem.com/opensource/ahk-callbox.ahk>

Mise en oeuvre

1. Téléchargez l'exécutable depuis <http://mir.thinkrosystem.com/opensource/ahk-callbox.exe>
2. Configurez le poste informatique afin qu'il lance cet exécutable au démarrage.

Utilisation

Pour lancer un appel depuis une sélection :

1. Sélectionnez un numéro de téléphone (avec la souris par exemple).
2. Réalisez la combinaison de touches [SHIFT]+[CTRL]+[P]



5 Liens tel : callto : et callbox :

Le lancement d'appel via liens de type tel : callto : et callbox : fonctionne sous Windows et macOS¹.

5.1 Mise en oeuvre d'un lien d'appel dans une page HTML

Des liens d'appel peuvent être mis en oeuvre (notamment au sein de web applications) pour obtenir la fonctionnalité de click-to-call de la façon suivante :

```
<a href="callbox:+33102030405">+33 (0)1 02 03 04 05</a>
```

1. Le fonctionnement des liens de lancement d'appel sous macOS est réalisé en tâche de fond par le service CallBoxCompanion installé et démarré automatiquement par l'application CallBox UI. Des autorisations supplémentaires dans les paramètres de sécurité de macOS peuvent être nécessaires pour un bon fonctionnement des liens d'appel.



6 API CallBox Web

6.1 Pré-requis

Configuration du système CallBox :

1. Démarrez le logiciel CallBox UI et rendez-vous dans la perspective "Configuration" → onglet "Paramètres" → section "Configuration API".
2. Activez l'API Callbox Web.

Sur le poste informatique de l'utilisateur :

1. Installez et démarrez le logiciel CallBox UI.
2. Rendez-vous dans la perspective "Supervision" du logiciel CallBox UI et identifiez-vous en tant qu'utilisateur CallBox.

6.2 Notification d'appel (remontée de fiche)

L'activation et les notifications d'appel sont configurables depuis le logiciel CallBox UI, dans la perspective "Configuration" → onglet "Paramètres" → section "Configuration API".

Options de configuration de CTI Web

- Configuration de l'URL à ouvrir lors d'une activation.
- Activation sur sonnerie d'un appel entrant.
- Activation sur sonnerie d'un appel sortant.
- Activation sur décroché d'un appel entrant.
- Activation sur décroché d'un appel sortant.

Fonctionnement

Lors de l'activation, l'URL configurée est ouverte dans le navigateur par défaut du poste et la chaîne ##NUM## est remplacée par le numéro de téléphone de l'appelant.

6.3 Lancement d'appel (click-to-call)

6.3.1 Web application

Mise en oeuvre

1. Insérer un iframe dans la page web d'où l'appel téléphonique doit être lancé :

```
<iframe style="display:none;" name="hiddenframe">
```

2. Afin de lancer l'appel téléphonique, ajouter un lien de la forme suivante :

```
<a href="http://localhost:3008/0606060606" target="hiddenframe">Appeler</a>
```



7 API CallBox via exécutable

Pour plus d'informations concernant l'utilisation de l'API CallBox via un exécutable, référez-vous au manuel utilisateur CallBox.



8 Contrôle d'un élément de routage "Switch App"

L'élément de routage Switch App permet de contrôler l'aiguillage des appels entrants vers différentes routes (ou branches de routage d'appel) via un web service : lorsqu'un appel entrant qui chemine dans le routage d'appel traverse un Switch App, celui-ci envoie les informations de cet appel vers l'URL du web service définie dans la configuration du Switch App et reçoit du web service le numéro de la route du Switch App vers lequel diriger l'appel.

Note : référez-vous au manuel utilisateur CallBox pour la création et la configuration d'un Switch App.

Mise en oeuvre d'un web service de contrôle d'un Switch App

Lorsqu'un appel entrant traverse le Switch App à contrôler, le serveur CallBox émet une requête HTTP vers l'URL définie dans le Switch App avec le paramètre POST nommé "query" contenant le JSON suivant :

```
{
  "callerid": "NUMERO APPELANT",
  "callername": "NOM APPELANT",
  [Optionnel] "dtmf": "VALEUR RECUEILLIE VIA DTMF",
  [Optionnel] "stt": "VALEUR RECUEILLIE VIA RECONNAISSANCE VOCALE"
}
```

Le serveur CallBox attend en retour le numéro de la route vers laquelle router l'appel en JSON :

```
{
  "goto": "NUMERO DE LA ROUTE"
}
```

Sans réponse de la part du web service dans le temps imparti (1 seconde), le Switch App dirigera l'appel vers la route numéro 0 "Par défaut".



9 Import automatique de contacts

Il est possible d'importer automatiquement des contacts depuis plusieurs sources simultanément. Chaque source de contacts est importée dans un groupe distinct de contacts (en lecture seule) utilisable dynamiquement dans le routage d'appel.

9.1 Fichier CSV

Si les contacts sont répartis dans plusieurs groupes, prévoyez un fichier par groupe.

9.1.1 Structure du fichier

- Format du fichier : CSV.
- Jeu de caractères (*charset*) : UTF-8.
- Séparateur de champs : point-virgule [;].
- Aucun encadrement du contenu des champs par des caractères spéciaux ou contenu des champs encadré par des guillemets (*double quote*).
- Pas de première ligne contenant les entêtes des champs.

Contenu du fichier :

- Champ "Type" : un chiffre de 1 à 5 qui détermine le type de contact.
 - 1 = Téléphone fixe.
 - 2 = Téléphone portable.
 - 3 = Fax.
 - 4 = Adresse email.
 - 5 = N° poste CallBox.
- Champ "Nom du Contact" : chaîne au format texte.
- Champ "Etiquette" : chaîne au format texte (peut être vide).
- Champ "Coordonnée" : chaîne au format texte (afin de préserver le zéro en début de chaîne, le cas échéant). Les numéros de téléphones ne doivent comporter aucun séparateur.

Note : toute ligne non conforme ne sera pas importée.

Exemple de fichier de Contacts

Un fichier de test au format CSV est disponible à l'adresse suivante :

<https://mir.thinkrosystem.com/test/CallBoxContactImport.csv>

Il est conseillé d'effectuer vos tests avec ce fichier d'exemple.

9.1.2 Récupération automatique des données

Mise en oeuvre

Depuis l'application CallBox UI, rendez-vous dans la perspective Carnet d'adresses :

1. Depuis le sous-menu **+**, cliquez sur **“+ Ajouter un groupe import auto”**.
2. Nommez le groupe et saisissez l'URL du fichier contenant les contacts à importer.



3. L'import démarre immédiatement (l'opération peut prendre plusieurs minutes en fonction de la quantité de données à importer).

Si vous créez plusieurs groupes d'import auto en même temps, attendez que les données du premier groupe créé soient importées en totalité avant de créer le groupe suivant.

Actualisation des données des contacts

Toutes les heures, le serveur CallBox téléchargera automatiquement le fichier CSV via HTTP à l'adresse définie lors de la création du groupe d'import auto.

Lors de l'import des contacts contenus dans le fichier CSV, les contacts présents au sein du groupe de destination seront remplacés par les contacts importés.

En cas d'échec de l'import automatique, les données du dernier import ayant fonctionné sont conservées en cache et un email de notification est automatiquement envoyé pour signaler le dysfonctionnement.

9.2 Contacts Gmail™

Vous pouvez importer automatiquement vos contacts Gmail™ après une simple procédure d'autorisation auprès de Google.

9.3 A la demande

Si vous souhaitez importer des contacts depuis votre application métier ou depuis toute autre source de contacts, [contactez Thinkro System](#).



10 Contacts partagés

Le système CallBox permet de récupérer les contacts depuis une application cliente de contacts ou un téléphone au format CardDAV ou LDAP.

10.1 Contacts partagés CardDAV

Configuration du compte CardDAV

- URL du compte CardDAV :

```
https://[IP-SERVEUR-CALLBOX]:[PORT-ADMINISTRATION]/callbox/carddav/server.php/addressbooks/
```

- Identifiant CardDAV : admin
- Mot de passe CardDAV (par défaut) : *consultable et modifiable depuis l'Espace Installateur CallBox*

10.1.1 Correspondance Champs CardDAV / Contacts CallBox

- Champ *Name* (N) :
 - Family Names* : "Nom du contact"
 - Given Name* : "Etiquette"
 - Additional Names* : "Groupe du contact"
- Champ *Formatted Name* (FN) : "Nom du contact" + "Etiquette"
- Champ *Telephone* (TEL) : "Téléphone fixe" (contact type 1), "Téléphone mobile" (contact type 2), "N° poste CallBox" (contact type 5) ou "Fax" (contact type 3)
- Champ *Email* (EMAIL) : "Email" (contact type 4)

10.2 Contacts partagés LDAP(S)

L'intérêt du carnet d'adresses LDAP ou LDAPS est notamment de partager les entrées du carnet d'adresses CallBox avec des matériels téléphoniques compatibles avec le système Callbox mais qui ne sont pas certifiés "CallBox Full Features".

Il présente également l'avantage de se défaire de la limitation du nombre d'entrées des carnets d'adresses des téléphones lorsque c'est indispensable.

10.2.1 Configuration réseau

Configurez l'un des transferts de ports sur le modem/routeur du réseau hébergeant le serveur CallBox tel qu'indiqué dans le Guide d'Installation.

L'accès au carnet d'adresses LDAP est restreint au réseau local (LAN) et aux IPs d'administration autorisées.



10.2.2 Paramètres LDAP(S) CallBox

- Base DN (*"Distinguished Name"*)² : `dc=callbox,dc=network`
- Identifiant (*"DN user"*, *"User Name"* ou *"User ID"*) : `cn=root,dc=callbox,dc=network`
- Mot de passe : consultable dans l'Espace Installateur CallBox

10.2.3 Correspondance Champs LDAP(S) / Contacts CallBox

- Common Name (cn) : "Groupe du contact" + "Nom du contact" + "Etiquette"
- Given Name (gn) : "Nom du contact"
- Surname (sn) : "Etiquette"
- Organisation (o) : "Groupe du contact"
- Data :
 - mail : "Email"
 - telephoneNumber : "Téléphone fixe", "Téléphone mobile" ou "N° poste CallBox"
 - facsimileTelephoneNumber : "Fax"

2. Pour se limiter à un groupe de contacts particulier : `o=NOM DU GROUPE,dc=callbox,dc=network`

