

# Le CTI et le développement TAO avec Genesys et Alcatel

par [Cédric Chatelain](#)

Date de publication : 15/12/2005

Dernière mise à jour :

Le bandeau CTI (ou bandeau de TAO) est un des points clé du centre d'appels utilisant le CTI. A la fois IHM et point de liaison entre la téléphonie, le système d'informations et l'utilisateur il ne faut pas le négliger. A développer en interne ou à intégrer sous forme de connecteur, tout dépend : des besoins, de la façon de travailler des télé conseillers, des types d'appels à traiter et du type de système téléphonique. Pour ma part je vais aborder ici la TAO avec serveurs Genesys, serveurs CTI avec lesquels je travaille. Pour rappel, certains termes techniques utilisés ici ont été présentés dans les articles Les centres d'appels et le CTI et Présentation de la distribution des appels sur Alcatel.

- 1 - Qu'est-ce qu'un bandeau de TAO et à quoi sert il ?
- 2 - Développement interne et/ou connecteur ?
  - 2.1 - Introduction
  - 2.2 - Le choix de TAO pour une architecture CRM web du type Siebel
- 3 - Les bases dans le développement TAO
  - 3.1 - La connexion au TServer
  - 3.2 - Vérifier l'enchaînement des évènements téléphoniques et leur disponibilité sur le CTI
- 4 - Traiter les évènements et méthodes de base
  - 4.1 - La connexion au TServer
  - 4.2 - Login et logout d'un agent
  - 4.3 - Passage en Prêt / Retrait / Post traitement
  - 4.4 - Exemple de code de connexion au TServer et login
- 5 - Gérer les appels
  - 5.1 - Introduction
  - 5.2 - Déterminer le type d'appels
  - 5.3 - Les appels ACD
  - 5.4 - Les appels sortants
  - 5.5 - Les transferts d'appels
  - 5.6 - Exemple de squelette Delphi de traitement de l'évènement established
- 6 - Gérer les données attachées et les user events
  - 6.1 - Les données attachées
    - 6.1.1 - Récupérer des données attachées
    - 6.1.2 - Ajouter des données attachées
  - 6.2 - User events
    - 6.2.1 - Envoyer un user event

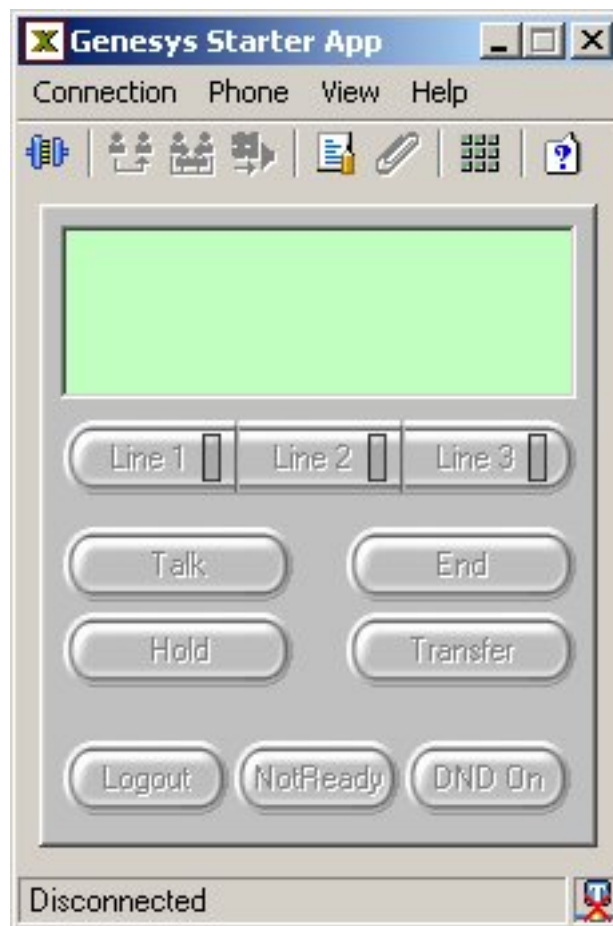
## 1 - Qu'est-ce qu'un bandeau de TAO et à quoi sert il ?

TAO c'est Téléphonie Assistée par Ordinateur. Le bandeau de TAO est donc l'interface informatique du téléphone. Elle a pour but :

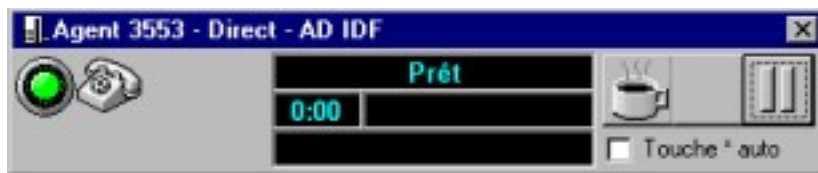
- de permettre au télé conseiller de manipuler son téléphone via son PC (décrocher pour prendre un appel, mettre un appel en attente, effectuer un transfert, un appel de consultation, une conférence ou raccrocher un appel)
- de fournir à un logiciel (de vente ou de gestion de relation clients) des informations sur les appels afin d'ouvrir automatiquement une fiche clients par exemple.
- de recevoir des informations de ce même logiciel dans le cadre de planification d'appel ou de rappels.

Le bandeau de TAO est donc un des points importants du centre d'appels. Bien pensé et optimisé il permet un gain de temps au début des appels sur l'ouverture de la fiche du client.

Ce temps généralement mesuré en secondes peut paraître minime, mais lorsque l'on attend au téléphone, que l'on a déjà patienté avant de pouvoir parler à quelqu'un, même une seconde a son importance.



*exemple de bandeau de TAO fourni par Genesys*

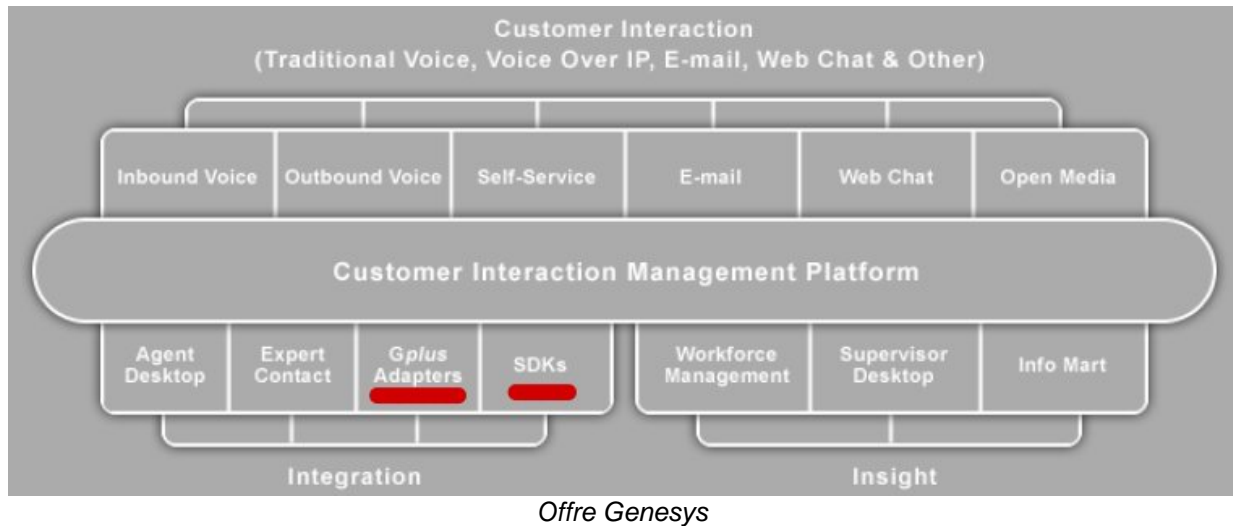


*exemple de bandeau développé en interne*

## 2 - Développement interne et/ou connecteur ?

### 2.1 - Introduction

Genesys propose entre autre un SDK et un connecteur pour réaliser l'intégration des fonctions CTI.



Offre Genesys

Quel plaisir quand on a le choix. Mais parfois il est dicté par la technologie du logiciel de vente / gestion de la relation clientèle.

Quand j'ai commencé à travailler sur le CTI il s'agissait de faire un couplage Genesys / Vantive. Vantive fonctionnait (le produit a disparu) sur une technologie client serveur avec un client lourd. Les deux solutions étaient donc aisées à mettre en place. Comme à cette époque (c'était en 1998) le CTI commençait à faire son apparition, les connecteurs n'étaient pas encore jugés assez complets et c'est le choix d'un développement Delphi qui a été fait.

Il y a deux ans il a été décidé d'abandonner Vantive au profit de Siebel. La question développement ou connecteur s'est donc à nouveau posée. Mais cette fois ci il a bien fallu tenir compte de l'historique, les conseillers étant habitués à l'ergonomie du bandeau de TAO et nous à une certaine souplesse dans le traitement des données CTI.

### 2.2 - Le choix de TAO pour une architecture CRM web du type Siebel

Dans un tel contexte il est naturel de commencer par regarder le connecteur Genesys / Siebel. L'intégration des fonctionnalités CTI entre un bandeau de TAO stand-alone et un browser web utilisé en client léger devient en effet vite assez complexe à étudier et il y a de forts risques que cela donne naissance à une véritable "usine à gaz" ne serait ce que pour les fonctions de base du couplage téléphonie informatique.

Mais si l'utilisation du connecteur pour assurer les échanges Genesys / Siebel a semblé s'imposer de lui-même il nous restait quelques points sur lesquelles on se posait des questions :

- l'ergonomie du connecteur ne nous satisfaisait pas à 100%
- le traitement d'évènements et/ou de données téléphoniques spécifiques à notre centre d'appels étaient impossible

Il faut en effet savoir que les connecteurs sont en plein développement. La particularité de ceux disponibles pour Siebel / Genesys (en tous cas pour les versions qui étaient disponibles en 2004) est de gommer les singularités des PABX utilisés. Donc seules les fonctionnalités 100% standard sont supportées, ce qui nous gênait dans une certaine mesure pour exploiter pleinement les fonctionnalités de notre PABX Alcatel.

La solution retenue a donc consisté en un mix connecteur / développement interne. Le connecteur a été mis en place afin de permettre une transmission aisée des événements CTI de base à Siebel (début et fin d'appels, connexion et déconnexion des agents etc etc...). Alors que le bandeau préexistant et développé en Delphi a pris en charge la partie IHM avec l'affichage des données spécifique à notre centre d'appels, la gestion de fonctionnalités téléphoniques spécifiques à Alcatel ainsi que le formatage de données attachées à l'appel.

Cette solution nous a été d'autant plus profitable qu'une partie fonctionnelle spécifique à nos besoins est toujours traitée dans un bandeau de TAO développé entièrement en interne. Ces traitements auraient été possibles en faisant des scripts dans Siebel mais nous y avons vu plusieurs inconvénients :

- On se serait écarté des standards Siebel pour l'intégration Genesys et cela aurait compliqué les travaux de montée de version
- Les évolutions CTI (dont on sait sur nos plateaux d'appels qu'elles doivent être traitées rapidement) auraient été ralenties par la complexité des procédures de développement et de mise en production Siebel (qui a des contraintes de projet très fortes chez nous).

## 3 - Les bases dans le développement TAO

### 3.1 - La connexion au TServer

Entre les versions 5 et 7 de Genesys on note l'accent mis sur la haute disponibilité des serveurs de CTI mais aussi sur la possibilité de gérer des centres d'appels en multi-site. Pour y parvenir Genesys gère le TServer (pièce principale de la solution CTI Genesys) ainsi que les autres modules qui viennent s'y connecter dans niveau du serveur de configuration.

En toute logique le bandeau de TAO en fera autant, il ne se connectera pas directement au TServer mais passera par le serveur de configuration et grâce à un nom d'application qui lui sera propre déterminera le TServer qui sera utilisé et sur quel port. Cela permettra d'obtenir les avantages suivants :

- il suivra la gestion haute disponibilité si celle-ci a été mise en place sur les serveurs
- les changements d'installation des différents TServer pourront être effectués sans se soucier de changer quoi que ce soit au niveau de la configuration / du développement des bandeaux de TAO
- La gestion des différents environnements (développement, recette, production) et des différents sites de production en est grandement facilitée.

Dans le but de profiter pleinement des avantages décrits ci-dessus il faut, bien sûr, prévoir à l'avance les différents environnements / sites d'appels qui seront mis en place ainsi que les configurations des différentes instances du bandeau de TAO à déclarer.

*Il est à noter que si cela est possible en développement interne, ce fonctionnement n'est pas possible avec le connecteur Siebel qui lui doit se connecter en direct sur un TServer. Du moins pas avec les versions 6.X du connecteur Genesys.*

### 3.2 - Vérifier l'enchaînement des événements téléphoniques et leur disponibilité sur le CTI

Là aussi cela peut paraître basique comme vérification, mais il est nécessaire de vérifier l'ordre des événements envoyés par le PABX et de voir si on peut les gérer par le CTI.

On peut prendre un exemple concret avec les événements de fin d'appels sur Alcatel. Quand un appel distribué par l'ACD prend fin le PABX fait passer l'agent par les phases suivantes :

- raccrochage de l'appel
- passage en mode de post-traitement
- passage en "pause syndicale" (si celle-ci est activée sur le PABX)
- passage en mode prêt ou retrait en fonction de la configuration du groupe de travail

La réception de ces événements ne pose aucun problème à Genesys. Ces phases peuvent être écourtées sur le téléphone sans que cela ne pose le moindre problème du côté du CTI. Mais imaginons que l'on veuille gérer la durée de chacune de ces phases via le bandeau de TAO.

Via le Desktop toolkit de Genesys nous ne pouvons ni demander la fin d'une phase en cours, ni commander le passage en "pause syndicale". La demande de passage en post-traitement, en mode prêt ou retrait est possible, mais pas en "pause syndicale". Cette notion peut être émulée sur le bandeau TAO alors que le téléphone est dans un mode qui est fonctionnellement semblable.

De tels cas sont heureusement rares mais il est de toutes façons recommandé d'avoir une bonne connaissance des fonctionnalités utilisées sur le centre d'appels et de s'assurer de ce qui est faisable en CTI avant de commencer l'étude du bandeau de TAO. du bandeau de TAO

## 4 - Traiter les évènements et méthodes de base

### 4.1 - La connexion au TServer

C'est la base de la gestion du bandeau de TAO. En réponse à la demande de connexion le TServer renvoie la liste des fonctionnalités supportées par le PABX auquel il est lui même connecté. Ce qui peut être utile quand on gère un centre d'appels disposant de plusieurs PABX différents.

### 4.2 - Login et logout d'un agent

Le login dépend du type d'agent et de la façon dont il travaille. Sur Alcatel il est possible de monitorer les postes de deux façons distinctes :

- on monitore un poste téléphonique sans qu'un agent ne soit connecté dessus.
- on monitore un poste sur lequel un agent se connecte

Le fonctionnement avec login agent est beaucoup plus souple en centre d'appels car il permet aux agents de se connecter sur n'importe quel poste et de changer de groupe de travail facilement. La distribution des appels et les groupes de travail sont présentés [ici](#).

Alcatel, par rapport à d'autres PABX présente une particularité : au moment où un agent se connecte le login de l'agent se substitue au numéro du poste téléphonique utilisé. Le résultat est un changement de numéro d'annuaire de la ressource en question. L'agent peut donc disposer d'une ligne directe qui a le même numéro quelque soit le poste de travail où il s'est installé.

*L'option "agent substitute" permet au bandeau de TAO de ne monitorer qu'une extension et de demander au TServer d'envoyer tous les évènements, même relatifs à l'agent, sur l'extension du poste téléphonique. Cette option est obligatoire pour le connecteur Siebel.*

### 4.3 - Passage en Prêt / Retrait / Post traitement

Le passage en prêt est simple à gérer. L'évènement reçu est un "TEventAgentReady". La demande de passage en prêt se fait avec la méthode "TReady". Alors que le passage en retrait ou post traitement est un peu plus subtil.

Les passage en retrait et en post traitement se font tous les deux sur réception d'un évènement "TEventAgentNotReady". Ce qui les distingue c'est le "WorkMode" reçu dans l'évènement. Ses valeurs possibles sont :

```
0 The work mode is unknown
1 The agent has to perform a manual operation to become available after a call is disconnected
2 The agent's availability is decided by the switch
3 If the agent work mode is set to AgentManualIn, this status specifies the condition assigned to the agent after the previous call has cleared and before the agent becomes available to receive the next call
4 The agent is not ready to receive calls
5 The agent work mode is set to NoCallDisconnect other than the default value
```

Dans le cas qui nous occupe ce sont les valeurs 3 et 4 qui nous intéressent. Elles seront aussi utilisées avec la méthode "TNotReady" qui nous servira à demander le passage en mode retrait ou en post traitement.

*Rappel : En mode prêt l'agent peut recevoir tous types d'appel (appel ACD / interne / direct sur son numéro), en mode retrait il ne peut recevoir que des appels internes ou directs et en post traitement il ne peut recevoir aucun appel.*

#### 4.4 - Exemple de code de connexion au TServer et login

Sur un PABX Alcatel un agent se connecte sur un poste (ou extension) et de vient un agent (ou position ACD) une fois connecté. Il est aussi possible d'utiliser le CTI directement pour prendre le contrôle d'un poste sans avoir besoin de se logger. C'est ce qui est géré dans cet exemple de code.

Dans cet exemple de code j'utilise une connexion classique au TServer, en configurant ce dernier en mode "agent-substitute" je n'aurais besoin de déclarer qu'une extension.

##### connexion au TServer et login

```

if ( AgentID = '' ) and ( AgentPwd = '' ) then
begin
  TExtension1.TExtensionType := TExtension2.TExtensionType;
  TExtension1.TDN := AgentDn;

  TitreBandeau := 'Poste ' + AgentDn;

end
else
begin
  TExtension1.TExtensionType := 2;
  TExtension2.TDN := AgentDn;
  TExtension2.TAgentID := AgentID;
  TExtension1.TDN := AgentID;
  TExtension1.TAgentID := AgentID;
  TExtension1.TAgentID := AgentID;
  TExtension1.TAgentPassword := AgentPwd;
  TExtension2.TAgentPassword := AgentPwd;

  AgentType := AGENT_ACD;

  TitreBandeau := 'Agent ' + AgentID + ' - ' + AgentServiceName + ' - ' + AgentAgenceName;

end;

case AgentType of
AGENT_ACD:
begin
  TExtension1.Enabled := true;
  if not TExtension1.Enabled then Tracer( 'Could not start Extension 1' );

  TExtension2.Enabled := true;
  if not TExtension2.Enabled then Tracer( 'Could not start Extension2' );

  TExtension1.TRegister;
  TExtension2.TRegister;

  TExtension2.TQueue := AgentQueue;
  TExtension1.TQueue := AgentQueue;

  // Le login se fait sur l'extension 2 (poste)
  // et c'est l'extension 1 qui sera ensuite utilisée.
  TExtension2.TLogin;
end;

AGENT_NON_ACD:
begin
  TExtension2.Enabled := false;
  if TExtension2.Enabled then Tracer( 'Could not stop Extension2' );

  TExtension1.Enabled := true;
  if not TExtension1.Enabled then Tracer( 'Could not start Extension1' );

  TExtension1.TRegister;
end;

```

connexion au TSrver et login

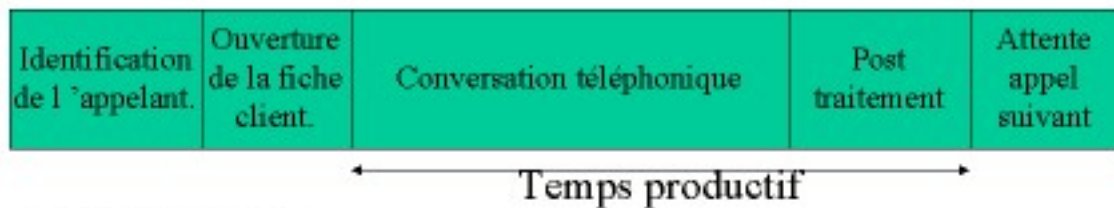
```
end; // Case
```

## 5 - Gérer les appels

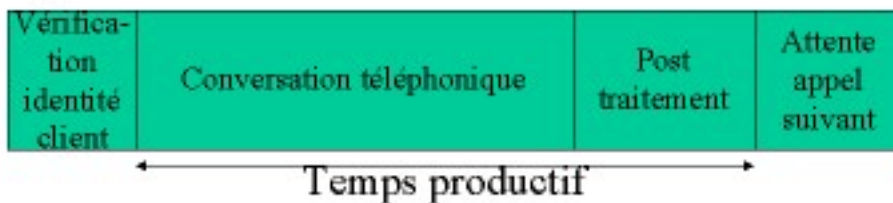
### 5.1 - Introduction

C'est le point de départ de la gestion des appels par le bandeau de TAO. C'est là que l'ouverture de fiche va être faite en automatique, et que le télé conseiller va commencer à gagner du temps.

#### Appels sans CTI



#### Appels avec CTI



*Gain de temps grace au CTI*

C'est au début de l'appel qu'on exploite toutes les données venant du PABX.

### 5.2 - Déterminer le type d'appels

La première chose à déterminer sur la prise d'appel est le type de cet appel :

- appel entrant sur un groupe d'agent
- appel entrant sur la ligne directe de l'agent
- appel interne
- appel sortant (manuel ou généré par le CTI)

Les évènements qui indiquent le début d'un appel sont "TEventRinging" au moment où le poste de l'agent sonne et "TEventEstablished" au moment où l'appel est effectivement transmis à l'agent.

La plupart du temps c'est le "TEventEstablished" qui est préféré pour traiter les données de début d'appel. Les informations à rechercher sont le "CallType" qui peut avoir les valeurs suivantes :

```
0 The type of this call is unknown
1 An internal call
2 An inbound call
3 An outbound call
4 A consultation call
```

Dans le cas d'un appel entrant il reste à déterminer si l'appel est arrivé sur la ligne directe de l'agent ou a été distribué par l'ACD. Cela déterminera s'il y aura une phase de post traitement en fin d'appel mais peut aussi influencer sur les possibilités de traitement en cours d'appel. Pour cela il suffit de vérifier l'attribut "FirstDNIS" de l'évènement de début d'appel. Il indique le point d'entrée dans le PABX. Si le numéro est celui d'un pilote c'est que l'appel est un appel ACD. Si c'est un numéro d'agent c'est que l'appel est un appel sur une ligne directe.

Ce n'est qu'une fois que l'appel aura été catégorisé que l'on pourra déterminer les traitements adéquates à appliquer, pour l'ouverture de la fiche du client mais aussi pour déterminer si et comment on va enregistrer des informations sur l'appel qui va avoir lieu. On pourra par exemple enregistrer les dates et heures d'appel du client et indiquer quel conseiller a traité la demande.

### 5.3 - Les appels ACD

C'est le type d'appels le plus courant. Un appel entrant est distribué par l'ACD et est traité par un agent. Les informations disponibles en début d'appel sont assez classiques :

- numéro de téléphone de l'appelant
- ressource d'entrée sur le PABX (le First DNIS dont que nous avons utilisé pour vérifier le type d'appel)
- temps d'attente avant distribution de l'appel

On peut dans certains cas aussi y trouver des données attachées. Les données transmises dans le champ SUU (aussi appelé mini message), par exemple, sont transmises au bandeau de TAO par ce biais.

### 5.4 - Les appels sortants

Nous verrons ici les appels sortants passés unitairement par l'agent. Les campagnes d'appels gérées en automatique seront abordées dans un autre article.

Il peut être pratique pour un attaché commercial d'avoir des outils facilitant l'appel d'un client. Par exemple, sur la fiche de celui-ci, un bouton qui déclencherait la composition du numéro.

Pour cela rien de plus facile : la méthode TDial appliquée au TGetDefaultCallObj de l'extension et le tour est joué...

De plus on interceptera l'évènement EventDialing pour traiter le cas où l'attaché commercial numérote directement sur son poste téléphonique.

### 5.5 - Les transferts d'appels

C'est le point le plus complexe dans la gestion des appels par TAO. Il faut gérer le transfert tant au niveau de l'agent qui va le provoquer que celui qui va recevoir le transfert. Il faut aussi distinguer 2 types de transfert : en un ou deux temps.

Le transfert en un temps est le plus facile à gérer : Ils se font sur des ressources telles que des pilotes dont on sait qu'elles peuvent accepter des appels sans avoir besoin de vérifier leur disponibilité.

Le transfert en deux temps commence par un appel de consultation. L'émetteur commence donc par faire un double appel pour consulter l'agent destinataire, puis soit il valide le transfert soit il reprend l'appel du client.

Dans les deux cas on aura pris soin d'attacher à l'appel les informations nécessaires pour ouvrir la fiche du client et positionner le logiciel de CRM sur l'écran approprié. Cela se fait en attachant des données à l'appel. Dans ce cas on appréciera les souplesses de la gestion des données attachées qui peuvent être disponibles lors de l'appel de consultation alors qu'elles ont été attachées sur l'appel du client.

### **consult-user-data**

**Default Value:** separate

### **Valid Values:**

**separate** Stores user data for original and consultation calls in separate structures

**inherited** Copies user data from an original call to a consultation call when the consultation call is established; thereafter, stores user data separately for the original and the consultation call

**joint** Stores user data for an original call and a consultation call in one structure

L'agent destinataire souhaité du transfert peut donc disposer des données relatives au client dès le début de l'appel de consultation. Cela peut l'aider à décider s'il va accepter ou non le transfert.

Côté initiateur du transfert la complexité réside dans la gestion de début et de fin de l'appel de consultation. On peut avoir ces enchaînements :

### **Dans le cas où le transfert d'appel est finalisé**

- début d'appel ACD
- composition de l'appel de consultation
- établissement de l'appel de consultation
- validation du transfert
- fin de l'appel de consultation
- fin de l'appel ACD

### **Dans le cas où l'appel de consultation n'aboutit pas**

- début d'appel ACD
- échec de l'appel de consultation
- reprise de l'appel ACD
- suite du traitement de l'appel

**Dans le cas où le destinataire ne souhaite pas accepter le transfert**

- début d'appel ACD
- composition de l'appel de consultation
- établissement de l'appel de consultation
- abandon de l'appel de consultation
- fin de l'appel de consultation
- reprise de l'appel ACD
- suite du traitement de l'appel

**5.6 - Exemple de squelette Delphi de traitement de l'évènement established**

## Traiter la prise d'appels

```

procedure TExemple.TExtension1TEventEstablished(Sender: TObject; var EventInfo: ITEventInfo);
// DESCRIPTION : Gestion de l'evenement established - traitement des appels
begin
    // TransferState est initialisée à TRANSFER_INIT lors de la composition d'un appel
    // de consultation en vue d'un transfert
    if TransferState <> TRANSFER_INIT then
        TransferState := TRANSFER_NONE;

    if EventInfo.CallType <> CALL_CONSULTATION then
        begin
            // On stocke le ConnID Genesys pour identifier l'appel en cours
            CallID := Event.ConnID.GetConnIDAsStr;
            label_ANI.Caption := Event.OtherDN;

            // si on trouve des données attachées pour un transfert d'appels
            if attached_data_transfert >= 0 then
                begin
                    // traitement d'appel en transfert (ouverture de fiche...)
                end
            else
                begin
                    case Event.CallType of
                    begin
                        CALL_INBOUND:
                            begin
                                // traitement d'appel entrant (ouverture de fiche...)
                            end;
                        CALL_OUTBOUND:
                            begin
                                // appel sortant
                                // traitement d'appel sortant (ouverture de fiche...)
                            end;
                        CALL_INTERNAL:
                            begin
                                // appel interne
                                // traitement d'appel interne
                            end;
                    end;
                end;
            end;
        end
    else
        begin
            if TransferState = TRANSFER_INIT then
                // si un appel de consultation a été fait en vue d'un transfert

```

### Traiter la prise d'appels

```
begin
  Tracer( 'Consultation call received' );
  // on traite ici l'appel de consultation du point de vue
  // de l'initiateur du transfert
end
else
begin
  // on traite ici l'appel de consultation du point de vue
  // du destinataire du transfert
end;
end;
end;
```

## 6 - Gérer les données attachées et les user events

### 6.1 - Les données attachées

Ce sont des informations stockées dans la mémoire du serveur CTI et qui sont associées à un appel ou à un évènement CTI. Elles permettent :

- de transmettre des données entre applications, entre applis de Gestion de Relation Clients
- de transmettre des données entre bandeau de TAO et
- de faire des statistiques CTI enrichies par des données clients

#### 6.1.1 - Récupérer des données attachées

Les données attachées se présentent sous la forme clé / valeur avec une zone pour préciser de quel type est la valeur : chaîne / numérique / binaire / liste. On y accède séquentiellement.

##### Récupérer des données attachées

```
NbKey := UsrData.GetCount;
i := 0;
while i < NbKey do
begin
  UsrPair := UsrData.Get(i);
  if UsrPair.type = CKVTypeString then DataString := UsrPair.StringValue;
  if UsrPair.type = CKVTypeNum then DataNum := UsrPair.NumValue;
  if UsrPair.type = CKVTypeBinary then DataString := UsrPair.BinaryValue;
  if UsrPair.type = CKVTypeList then DataString := UsrPair.ListValue;

  Tracer( 'User Data : ' + UsrPair.Key + ' = ' + UsrPair.StringValue );

  i := i + 1;
end;
```

#### 6.1.2 - Ajouter des données attachées

L'ajout des données se fait aussi séquentiellement. Toutefois il est possible de choisir si les données seront ajoutées en début ou en fin de pile.

##### Ajouter des données attachées

```
UsrPair := CoCTKVPair.Create;

UsrData.Clear;
UsrPair.Key := 'XXXXXXXXX';

UsrPair.Type_ := CKVTypeString;
UsrPair.NumValue := 'XXXXXXXXXXXXX';
UsrData.AddHead(UsrPair);

UsrData.Clear;
UsrPair.Key := 'XXXXXXXXX';

UsrPair.Type_ := CKVTypeNum;
UsrPair.NumValue := 999999999;
UsrData.AddHead(UsrPair);

UsrData.Clear;
UsrPair.Key := 'XXXXXXXXX';

UsrPair.Type_ := CKVTypeBinary;
```

### Ajouter des données attachées

```

UsrPair.BinaryValue := 1;
UsrData.AddTail(UsrPair);

UsrData.Clear;
UsrPair.Key := 'XXXXXXXXX';

UsrPair.Type_ := CKVTypeList;
UsrPair.ListValue := datalist;
UsrData.AddTail(UsrPair);

```

## 6.2 - User events

Ce sont des évènements non téléphoniques servant à échanger des données entre applications TAO et serveurs CTI. Les user events se différencient les uns des autres par leur destination et les données qui leur sont attachées. Un user event permet d'échanger des données attachées en dehors de tout contexte d'appel.

### 6.2.1 - Envoyer un user event

L'envoi d'un user event se fait sur une extension, qu'il y ait ou non un appel en cours n'a pas d'incidence.

#### Envoyer un user event

```

var
  l_UserData : CTKVList;
  l_UserPair : CTKVPair;
  l_UserEvent : TEventInfo;
begin
  if TExtension1.Enabled then
    begin
      // Caractéristiques de l'événement
      l_UserEvent := CoTEventInfo.Create;
      l_UserEvent.ThisDN := AgentID;
      l_UserEvent.OtherDN := AgentID;
      l_UserEvent.AgentID := AgentID;

      // Attachement des données

      l_UserData := CoCTKVList.Create;
      l_UserData.Clear;
      l_UserPair := CoCTKVPair.Create;

      // On extrait les clés et les valeurs de la chaîne

      l_UserPair.Key := Key;
      l_UserPair.Type_ := CKVTypeString;
      l_UserPair.StringValue := TheData;
      l_UserData.AddHead(l_UserPair);

      l_UserEvent.UserData := l_UserData;

      // Envoi de l'événement
      TExtension1.TSendUserEvent(l_UserEvent);

      l_UserPair.free;
      l_UserData.free;
      l_UserEvent.free;
    end
  else
    Result := FALSE;
  end;
end;

```