

Afin d'implémenter un autre terminal de paiement, il suffit d'écrire une classe qui surclasse la classe abstraite MercatorPayTerm.PayTerm. Cela requiert une référence vers MercatorTunnel.dll.

La classe est définie comme suit :

```
public abstract class PayTerm : IDisposable
{
    public bool Enabled { get; set; }

    public abstract string Lib { get; }

    public abstract MercatorPayTerm.RetIsPaymentReadyForPayTerm
    GetRetIsPaymentReady(int i_paiem, double amount, object type_paiem, string
    id_dev_pieds, object dev, DataSet billingEngineDataSet, bool multipleCurrency,
    ApiLangues langue);

    public abstract string RequestDebit(double montant, int appl_id, string
    mode_lib, string acode, RequestDebitStopDelegate requestDebitStopDelegate);

    public abstract void ToolsMenu(IMenuElement parentMenu);
}
```

S'agissant d'une classe abstraite, toutes les propriétés et méthodes abstraites doivent être surchargées.

La méthode **GetRetIsPaymentReady** est essentielle car c'est elle qui détermine si un paiement doit être traité par ce terminal.

Si le terminal n'est pas concerné par ce paiement, alors il faut renvoyer null.

Si on souhaite que le terminal soit utilisé pour ce paiement, alors il faut renvoyer une instance de MercatorPayTerm.RetIsPaymentReadyForPayTerm, en veillant à mettre Enabled à true et en fixant le montant à payer dans Amount. En principe, les autres propriétés peuvent être ignorées. CzamType, CzamLib et Acode sont toutefois des paramètres qui seront passés à RequestDebit, respectivement à appl_id, mode_lib et acode.

La méthode **GetRetIsPaymentReady** doit aussi contenir toute logique qui détermine si le paiement est recevable. (Le paiement est-il positif ? Le paiement ne dépasse-t-il pas un maximum autorisé). Dans le cas où on souhaite indiquer à Mercator que le paiement n'est pas acceptable, il suffit de renvoyer un RetIsPaymentReadyForPayTerm avec Enabled à false (valeur par défaut).

La méthode **RequestDebit** est le cœur même de la classe. C'est cette méthode qui sera appelée par Mercator lors d'un paiement. Elle devra en définitive envoyer le « montant » au terminal de paiement. Elle doit renvoyer une chaîne non nulle en cas de succès (le contenu de la chaîne n'est pris en compte que pour le terminal ATOS-Worldline). Si elle renvoie null, le processus de paiement s'interrompt.

La propriété **Lib** doit simplement renvoyer le libellé associé à ce terminal.

```
public override string Lib
{
    get { return "Czam"; }
}
```

La méthode **ToolsMenu** permet de compléter le menu « Outils / Terminal Paiements ». Il est ainsi possible de spécifier des points de menus spécifiques à ce terminal de paiement. (Par exemple, des

tests de diagnostic supplémentaires, des consultations d'état du terminal, ...). Cette méthode peut ne contenir aucun code. Pour ajouter un menu, il suffit d'utiliser la méthode **Add** sur le `IMenuElement` parentMenu passé en paramètre. Cette méthode Add attend ces paramètres :

```
string text, EventHandler clickHandler, bool enabled
```

Pour instancier ce `PayTerm`, il faut le faire via le customizer **Main**. Ce terminal doit être chargé dans

- `MercatorUi.Globals.PayTerm`
- Ou `MercatorUi.Globals.PayTerm2`
- Ou `MercatorUi.Globals.PayTerm3`

Attention : si l'option CZAM est utilisée, il ne faut pas charger ce `PayTerm` dans `MercatorUi.Globals.PayTerm`, qui est toujours réservé pour le Czam.

Si dessous un customizer qui montre comment charger et décharger correctement un `PayTerm` sur mesures dans `MercatorUi.Globals.PayTerm2` :

```
using System;

namespace Main
{
    public class Customizer : MercatorUi.ICustomizers.IExec
    {
        public void Main(MercatorUi.ICustomizers.ExecAction Action)
        {
            if (Action == MercatorUi.ICustomizers.ExecAction.DossierOpen)
            {
                MercatorUi.Globals.PayTerm2 = new ...;
            }
            else if (Action == MercatorUi.ICustomizers.ExecAction.DossierClose)
            {
                if (MercatorUi.Globals.PayTerm2 != null)
                {
                    MercatorUi.Globals.PayTerm2.Dispose();
                    MercatorUi.Globals.PayTerm2 = null;
                }
            }
        }
    }
}
```