

# Convertisseur d'applications ORPHEE

Mai 2007 fre

<b>Chapitre</b>	<b>Page</b>
<b>1 Présentation</b>	<b>1/1</b>
1.1 Objectif du Convertisseur	1/1
1.2 Installation	1/1
1.3 Accès au Convertisseur	1/2
1.4 Cohabitation avec Orphée	1/4
1.4-1 Partage d'informations	1/4
1.4-2 Activation	1/4
<b>2 Procédure</b>	<b>2/1</b>
2.1 Description des étapes	2/1
2.2 Récupération d'applications Orphée	2/2
2.3 Sélection de l'application Orphée	2/3
2.3-1 Servroot local	2/4
2.3-2 Servroot déporté	2/4
2.4 Visualisation des éléments constitutifs de l'application	2/5
2.4-1 Saisie du fichier destinataire	2/6
2.4-2 Déclaration	2/7
2.4-3 Entités relais	2/9
2.5 Sélection des éléments	2/10
2.5-1 Déclaration	2/10
2.5-2 Entités Relais	2/11
2.6 Extraction et Analyse	2/12
2.7 Paramétrage de la conversion	2/13
2.7-1 Règles de conversion	2/13

<b>Chapitre</b>	<b>Page</b>
2.7-2 Paramétrage généraux	2/14
2.7-3 Paramétrage des variables	2/17
2.7-4 Limitations et restrictions	2/25
2.7-5 Paramétrage de la logique combinatoire	2/27
2.7-6 Paramétrage des boîtes fonctionnelles	2/27
<hr/>	
2.8 Mémorisation et Rappel des paramètres	2/35
<hr/>	
2.9 Conversion	2/37
2.9-1 Le bloc de contexte	2/37
2.9-2 Les éléments à configurer	2/37
2.9-3 Le résultat de conversion du code	2/38
2.9-4 le récapitulatif des données	2/40
<hr/>	
2.10 Tableau de correspondance	2/41
2.10-1 Tableau de correspondance des variables	2/41
2.10-2 Tableau de correspondance de la logique combinatoire traduisible	2/44
<hr/>	
<b>3 Guide d'utilisation</b>	<b>3/1</b>
<hr/>	
3.1 Synoptique	3/1
<hr/>	
3.2 Préparation préliminaire	3/2
<hr/>	
3.3 Récupération des variables uniquement	3/2
3.3-1 Sur une nouvelle application PL7	3/2
3.3-2 Sur une application PL7 existante	3/3
<hr/>	
3.4 Récupération des fonctions d'Automatismes et de logique combinatoire (entités + variables)	3/4
3.4-1 Sur une application PL7 vierge	3/4
3.4-2 Sur une application PL7 existante	3/5
<hr/>	
3.5 Récupération du maximum d'une application	3/5
<hr/>	

---

### 1.1 Objectif du Convertisseur

---

Le convertisseur Orphée se compose de 2 parties:

1. Le **Convertisseur Ladder** qui permet la récupération partielle des traitements combinatoires d'une Application Série 1000 (Entités programmes avec les Données associées) en garantissant un fonctionnement analogue sur ATS.  
Les traitements concernés sont exclusivement combinatoires et décrivent :
  - soit une logique combinatoire correspondant à une partie d'une ENTITE,
  - soit une suite d'ENTITES dont le comportement fonctionnel dans PL7 sera identique à son original S1000
2. Le **Convertisseur de Variables** qui permet de récupérer des variables d'une Application Orphée (S1000) avec leurs symboles et leurs commentaires.

Une fonction de réaffectation ( ou d'association) des variables concernées par la conversion, assiste l'utilisateur pour obtenir une organisation optimale de ces variables dans PL7.

---

### 1.2 Installation

---

Le convertisseur Orphée est un outil optionnel de PL7 junior. A ce titre, il peut être installé:

- soit en même temps que PL7. Dans ce cas il suffit de le sélectionner parmi les éléments proposés par la procédure d'installation
- soit après PL7. Il suffit de ne sélectionner que le convertisseur dans la liste des éléments à installer. La procédure contrôle la présence de PL7 Junior sur le PC.

Si PL7 Junior n'est pas présent, l'installation est refusée.

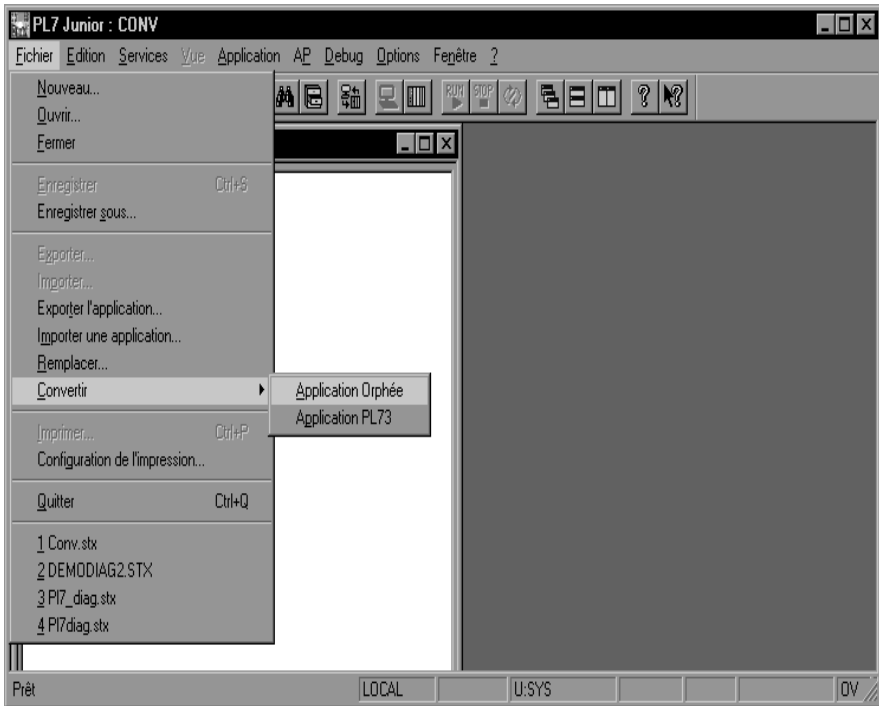
### 1.3 Accès au Convertisseur

Pour accéder au Convertisseur Orphée, il faut ouvrir une application PL7 qui peut être nouvelle ou existante. Puis accéder à l'item **Convertir** du menu **Fichier**. Cet item présente le choix de convertir une application PL7\_2, PL7\_3 ou Orphée selon le ou les convertisseurs installés.

#### Procédure:

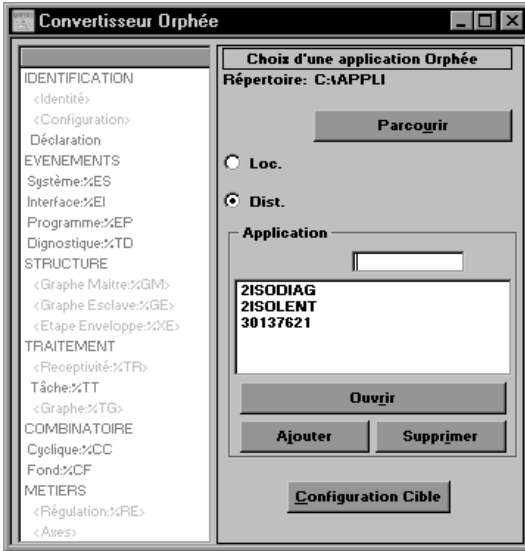
Pour activer le Convertisseur Orphée, il faut :

1. ouvrir une application PL7 (nouvelle ou existante),
2. sélectionner le menu **Fichier**,
3. choisir l'item **Convertir**,
4. choisir l'item **Application Orphée**.

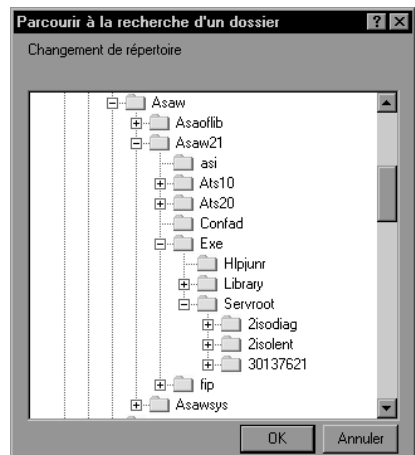


Après le lancement de l'outil de conversion d'une application Orphée, la fenêtre principale suivante s'ouvre.

Cette fenêtre permet à l'utilisateur de sélectionner une application Orphée (Loc.) locale (...\\SERVROOT).



Pour les applications se trouvant dans un autre répertoire (Dist.), un clic sur « Parcourir » ouvre l'écran de navigation ci-contre, permettant de sélectionner une application distante .



---

## **1.4 Cohabitation avec Orphée**

---

### **1.4-1 Partage d'informations**

L'exécution d'Orphée et du Convertisseur s'excluent mutuellement car ces 2 outils sont mono-utilisateur. Par exemple, si Orphée est déjà lancé le message suivant est alors affiché:

*"Le produit Convertisseur Orphée ne peut pas être lancé  
simultanément au produit ORPHEE".*

Orphée et PL7 peuvent s'exécuter en même temps. L'exclusion est faite seulement à l'activation du convertisseur.

Si Orphée est installé, le Convertisseur et Orphée partagent le même fichier «Orphée.ini» qui se trouve dans le répertoire de windows.

Dans le cas du "Servroot déporté"( cf § 2.3), ce fichier permet au convertisseur de partager avec Orphée le même espace d'installation des applications.

Dans le cas du "Servroot local", chaque outil a son espace propre d'installation des applications.

---

### **1.4-2 Activation**

L'installation de l'atelier Orphée sur le même PC n'est pas nécessaire pour exécuter le convertisseur.

Le Convertisseur ne nécessite pas l'usage de dongle.

Il n'est pas nécessaire que l'application Orphée soit validée pour faire la conversion de l'application.

### 2.1 Description des étapes

Le chapitre 2 décrit les différents traitements du Convertisseur. L'enchaînement logique et dynamique est expliqué dans le chapitre 3.

Les différents traitements se répartissent en 4 grandes étapes:

1	Sélection des éléments à convertir
2	Extraction et Analyse
3	Configuration de la Conversion
4	Conversion

Lors de la sélection des éléments à convertir, l'utilisateur précise les parties de l'application qu'il veut récupérer.

L'étape d'extraction et d'analyse est intégralement assurée par le convertisseur. Elle consiste à récupérer les éléments sélectionnés de la base de données d'Orphée et à les analyser.

L'étape suivante invite l'utilisateur à réaffecter les variables concernées et à fournir d'autres informations nécessaires à la conversion.

La dernière étape correspond à la conversion proprement dite, c'est-à-dire à la génération de code et/ou de variables au format source PL7.

Le résultat de la conversion doit être importé via l'outil adéquat de PL7 (Editeur Ladder ou Editeur de variables), pour être visible dans l'application cible.

**Note** : Chaque page décrivant un traitement comporte en bas de page un cartouche mettant en exergue l'étape à laquelle il fait partie.

Ex: 

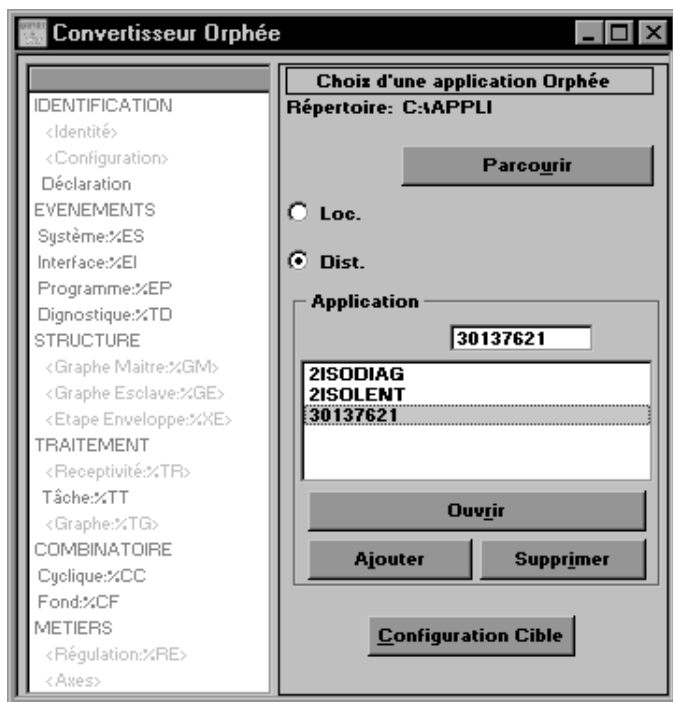
1 - Sélection des éléments à convertir
--

2	3	4
---	---	---



## 2.2 Récupération d'applications Orphée

Cette étape n'est pas obligatoire pour la récupération d'applications si celles-ci sont accessibles par le mécanisme de "servroot déporté". Cependant, le Convertisseur travaille plus rapidement avec son propre servroot. L'outil offre le moyen d'y inclure les applications. Il s'appuie pour cela sur le même service d'archivage et de restitution d'applications d'Orphée qui est accessible par le bouton "Ajouter"



Pour de plus amples informations sur ce service, consulter le manuel utilisateur d'Orphée.

**Rappel:** L'espace du servroot n'accepte que 16 applications au maximum.

Pour la gestion des applications de l'espace servroot, l'outil dispose du service "Supprimer" qui permet de détruire les applications devenues inutiles.

L'utilisation du "gestionnaire de fichier" Dos ou Windows n'est pas conseillée, car certains contrôles ne sont pas réalisés.

1 - Sélection des éléments à convertir

2

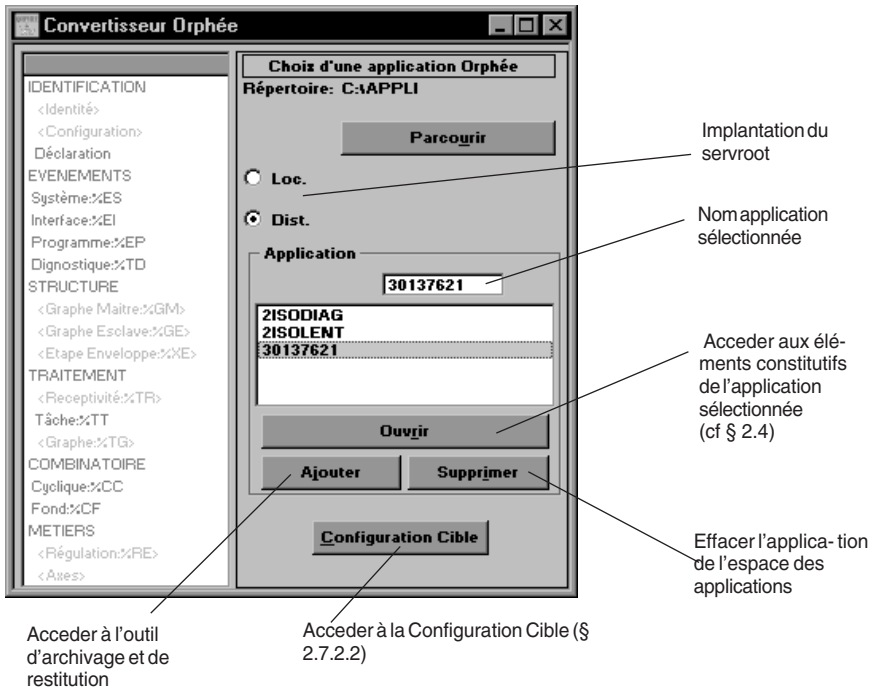
3

4

## 2.3 Sélection de l'application Orphée

La partie gauche présente la structure générique d'une application Orphée dont la barre de titre est destinée à contenir le nom de l'application Orphée sélectionnée.

La partie droite est réservée à la sélection de l'application Orphée de départ.



Cette fenêtre intègre trois services de gestion d'application **Ouvrir**, **Ajouter** et **Supprimer** avec un champ destiné à l'affichage du nom de l'application.

"Ouvrir" donne accès aux éléments constitutifs de l'application

"Ajouter" permet la restitution d'une application Orphée

"Supprimer" efface l'application du répertoire des applications

1 - Sélection des éléments à convertir

2

3

4

---

## Rappel préliminaire:

- 1 - On rappelle que le fichier ORPHEE.INI contient les informations exploitées dynamiquement par Orphée et le Convertisseur pour déterminer la localisation des applications.
2. Le servroot déporté est limité à 8 caractères.  
Tout chemin ne respectant pas cette contrainte n'est pas reconnu.  
Exemple: "c:\servd'.

---

### 2.3-1 Servroot local

Si ORPHEE.INI est absent ou s'il ne comporte pas la rubrique "Servroot=", le Convertisseur sélectionne automatiquement le répertoire SERVROOT "local" à PL7. Le bouton "**Loc**" est alors sélectionné.

Le chemin est présenté par la rubrique: "Répertoire:" (ASAWASAW21\EXE\SERVROOT dans notre exemple).

---

### 2.3-2 Servroot déporté

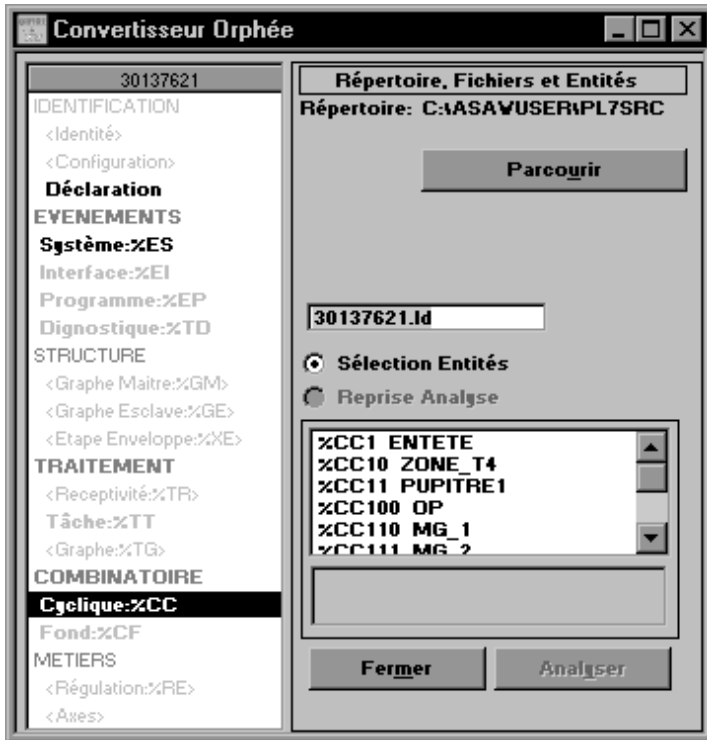
Si ORPHEE.INI existe et que la rubrique "Servroot=" est renseignée, le Convertisseur sélectionne automatiquement le chemin défini par la rubrique. Le bouton "**Dist.**" est alors sélectionné. Lorsque le bouton "**Dist.**" est sélectionné par une action volontaire de l'utilisateur, le Convertisseur présente une boîte de dialogue permettant de choisir le lecteur et le chemin d'accès.

L'information sur la localisation du servroot courant est sauvegardée dans le fichier ORPHEE.INI. L'espace peut se trouver sur n'importe quel lecteur accessible à partir du PC.

Si Orphée est activé entre deux sessions du Convertisseur, cette information peut être modifiée par l'utilisateur. C'est toujours le dernier servroot utilisé qui est mémorisé.

## 2.4 Visualisation des éléments constitutifs de l'application

Le service **Ouvrir** permet d'accéder au contenu de l'application Orphée. Il s'accompagne de la mise à jour de la structure de l'application. Les éléments programme "convertibles" figurent en caractères **gris** et les éléments programme que le Convertisseur ne prend pas en compte en <gris>.



Une fois l'application ouverte, la partie gauche de la fenêtre présente la structure générique de l'application avec en en-tête, un champ destiné à indiquer le nom de l'application sélectionnée.

Dans cette structure générique, les éléments constitutifs sont regroupés par "nature générique" : Déclaration pour les variables, %ES pour les Entités Evènements Système, %CC pour les Entités Combinatoires Cycliques, etc...

1 - Sélection des éléments à convertir	2	3	4
--	---	---	---

---

## 2.4-1 Saisie du fichier destinataire

Le fichier généré par la conversion a comme extension :

- **.ld** pour les entités relais,
- **.scy** pour les variables.

Il est stocké dans le "Répertoire des Sources" de PL7 (voir item "Personnaliser" du menu "Options" de PL7).

Par défaut, le Convertisseur propose le nom de l'application Orphée avec l'extension adéquate.

Il est possible d'effectuer plusieurs conversions (plusieurs passes) avant d'effectuer les opérations d'import, notamment pour les entités où les natures génériques ne peuvent pas être mélangées. Dans ce cas, il faut donner un nom de fichier différent pour chaque passe, car le Convertisseur écrase le contenu du fichier.

### Procédure :

Pour saisir le fichier destinataire :

- choisir le lecteur et sélectionner le répertoire via le bouton " Parcourir ",
- saisir le nom du fichier destinataire.

1 - Sélection des éléments à convertir

2

3

4

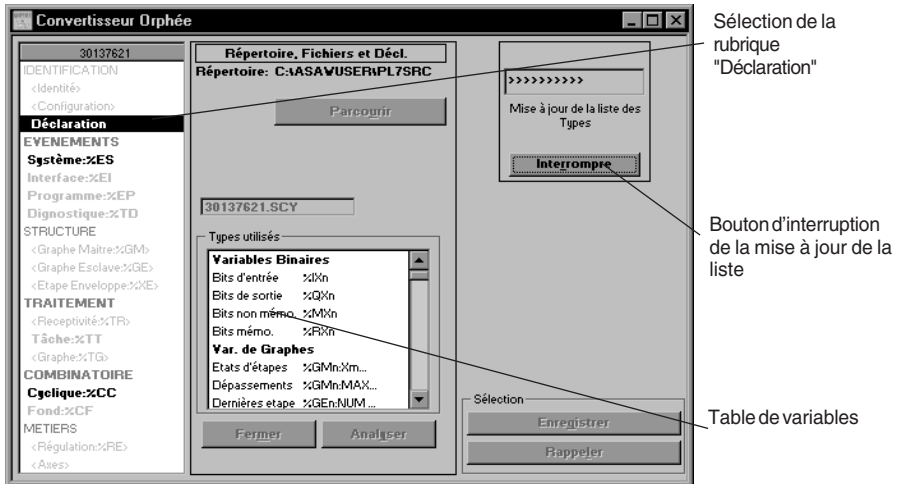
## 2.4-2 Déclaration

### Procédure :

Pour accéder à la liste des données, il faut :

- sélectionner la rubrique "Déclaration",
- attendre la fin de la mise à jour de la liste des types de variables. Cette opération est interruptible,
- saisir le nom du fichier qui contiendra les variables converties.

Les types de variables sont présentés dans une liste, regroupés par famille.



1 - Sélection des éléments à convertir

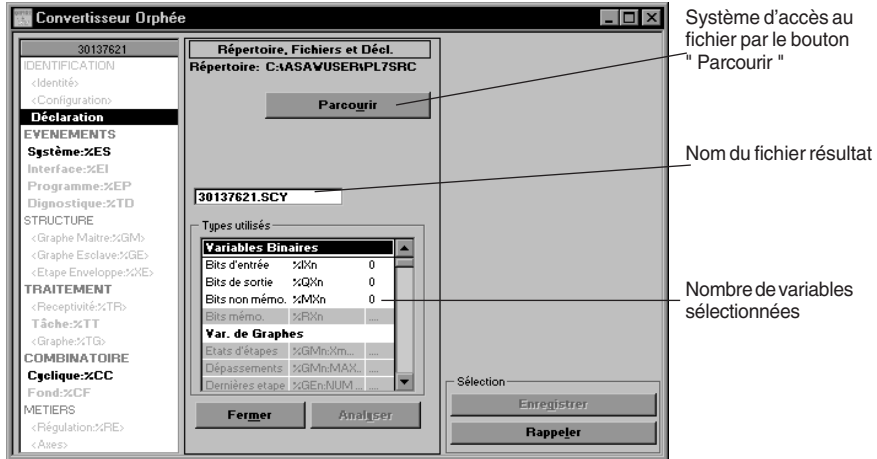
2

3

4

A la fin de la mise à jour, le Convertisseur affiche dans la partie centrale :

- le système d'accès au fichier résultat de la conversion (cf § 2.4.1 : saisie du fichier destinataire)
- la liste des types de variables regroupés par famille, avec le nombre de variables sélectionnées pour chaque type.



1 - Sélection des éléments à convertir

2

3

4

### 2.4-3 Entités relais

#### Procédure :

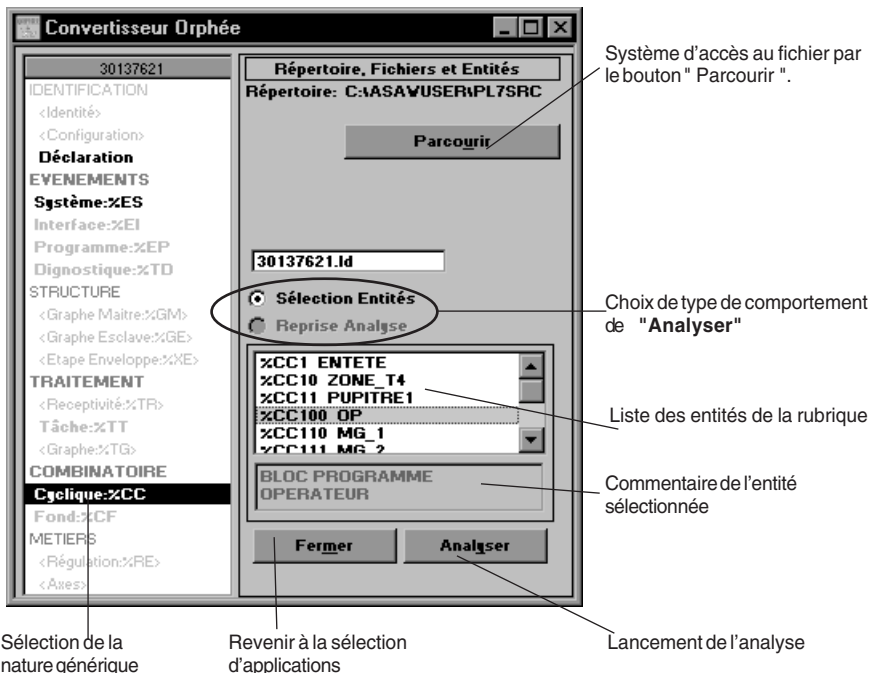
Pour accéder aux entités relais, il faut:

- sélectionner une nature générique (%ES, %CC, %CF, %TD,...),
- saisir le nom du fichier qui contiendra le code des entités sélectionnées et des variables qui y sont utilisées.

Le bouton **Analyser** a deux comportements:

sur **Sélection Entités** il extrait d'Orphée les entités sélectionnées, puis affiche les paramètres de conversion.

sur **Reprise analyse** il rafraîchit les paramètres de la conversion précédente.



Tant que l'analyse n'est pas demandée, il sera toujours possible de revenir à la liste des applications en sélectionnant le bouton **"Fermer"**.

1 - Sélection des éléments à convertir

2

3

4



## 2.5 Sélection des éléments

La sélection d'au moins une variable ou d'une entité relais donne accès aux boutons "Analyser" et "Enregistrer". La suite de l'analyse est décrite au § 2.6: Extraction et Analyse.

### 2.5-1 Déclaration

La sélection des variables se fait type par type.

Pour constituer la liste des variables à traduire, il faut:

- Sélectionner la rubrique "Déclaration".
- Choisir un type de variables (%IXn, %QXn, %RXn, ... ) dans la rubrique "Types utilisés". Le choix d'un type de variables entraîne l'affichage de toutes les variables de ce type utilisées dans l'application Orphée.
- Sélectionner les variables sur la partie droite. Le nombre de variables ainsi sélectionnées est affiché en temps réel dans la rubrique "Types utilisés", avec le type de variables correspondant.

**Convertisseur Orphée**

30137821

Répertoire, Fichiers et Décl.  
Répertoire: C:\ASAV\USER\PL7SRC  
Parcourir

Utilisés

DAPP	%MX0
PINIT	%MX10
MDEF	%MX11
IESG	%MX40
INSG	%MX60
ACSQA	%MX61
ACSQB	%MX62
ACSQC	%MX63
XPSQ1A	%MX61
XPSQ1B	%MX62
XPSQ1C	%MX63
MX39	%MX39
ICPN	%MX150
ICPN12	%MX151
IDVS12	%MX152
IDVS34	%MX153
IDVS66	%MX154

30137821.SCY

Types utilisés

Variables Binaires

Bits d'entrée	%IXn	0
Bits de sortie	%QXn	0
Bits non mémo.	%MXn	4
Bits mémo.	%RXn	

Var. de Graphes

États d'étapes	%GMn/M...
Dépassements	%GMn/MAX...
Dernières étapes	%GEN/NUM...

Fermer Analyser Enregistrer Rappeler

Sélection

1 - Sélection des éléments à convertir

2 3 4

Sélection de la rubrique **Déclaration**

Lancement de l'analyse

Revenir à la sélection de l'application

Liste des variables du type choisi

Nombre de variables sélectionnées

Mémorisation et rappel des sélections (cf. §2.8)

## 2.5-2 Entités Relais

La sélection des entités se fait dans une nature générique. On ne peut pas convertir simultanément des entités de natures génériques différentes.

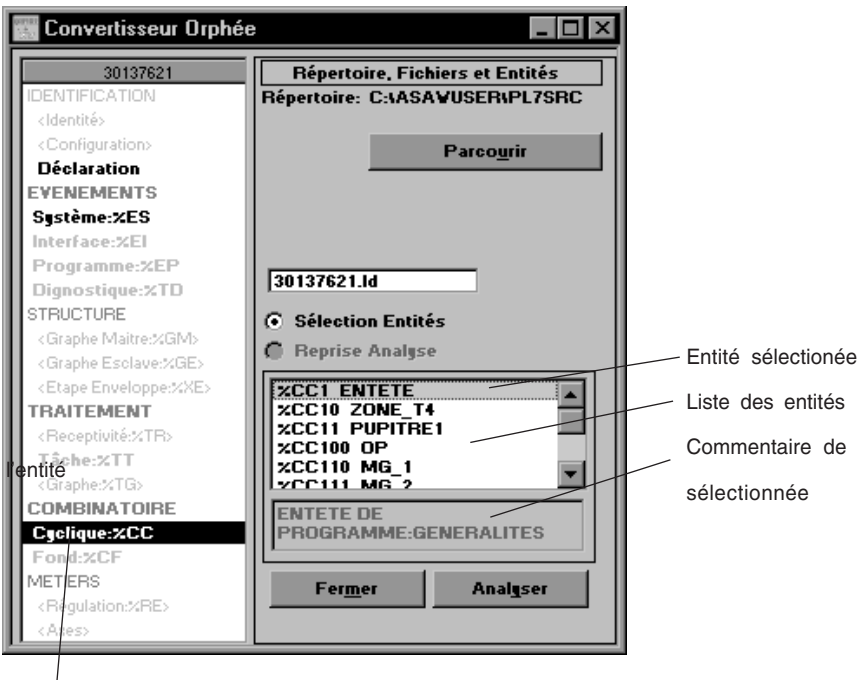
### Procédure :

Pour constituer la liste des entités relais à traduire, il faut :

- sélectionner une nature générique
- sélectionner les entités dans la liste affichée dans la partie droite

### Note.

Le commentaire de la dernière entité sélectionnée est affiché pour information.



Sélection d'une nature générique

Les natures génériques traductibles par le Convertisseur sont les suivantes :

- les évènements (%ESn, %EIn, %EPn, %TDn),
- les traitements (%TTn),
- les entités combinatoires (%CCn, %CFn).

1 - Sélection des éléments à convertir

2

3

4

---

## 2.6 Extraction et Analyse

---

Tous les éléments d'Orphée doivent être extraits de l'application avant d'être analysés par le Convertisseur.

Pour les variables, l'extraction est faite pendant la mise à jour des types

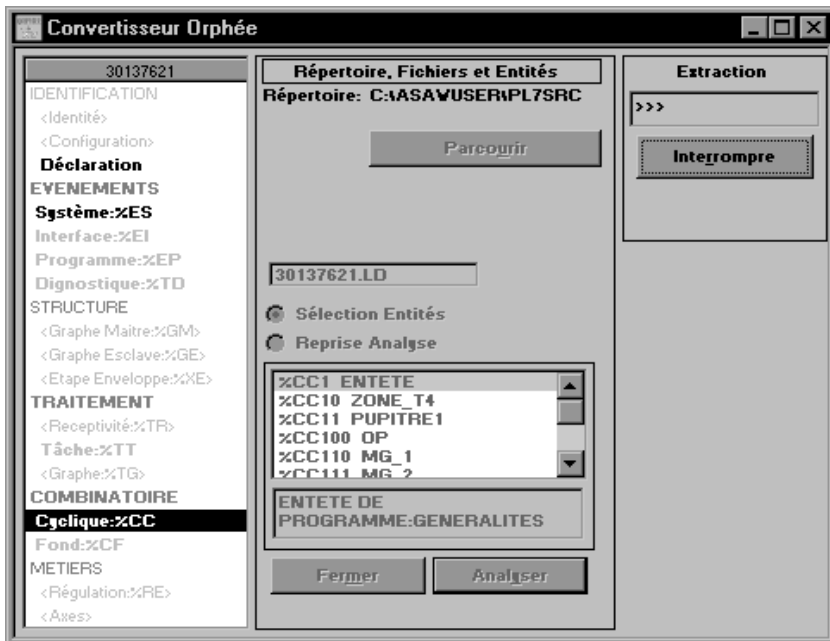
Pour les entités, elle est lancée automatiquement dès qu'on demande l'analyse.

L'extraction est visualisée par une jauge horizontale. Elle est interruptible. L'analyse fournit :

- les déclarations de l'application d'origine
- les indéterminations logiques,
- les indéterminations fonctionnelles.

### Procédure :

- cliquer sur le bouton Analyser, lorsque les éléments à convertir sont sélectionnés.



---

## 2.7 Paramétrage de la conversion

---

Après l'analyse, une fenêtre présente :

- **en caractères gras** : tout ce qui ne pose pas de problème de conversion.
- **en caractères "Rouge"** : tous les éléments qui demandent l'intervention ou l'attention de l'utilisateur, à savoir:
  - les éléments de programme impliqués dans la conversion pour lesquels il n'a pas été trouvé de correspondance dans la configuration logicielle ou matérielle de l'application cible,
  - les paramètres à saisir obligatoirement,
  - les variables en conflit avec l'application cible.
- **en caractères "grisés"** : tout ce qui n'est pas impacté par la conversion en cours.

---

### 2.7-1 Règles de conversion

#### Variables:

Voir Chapitre 2.10.1 pour la correspondance des variables.

#### Expressions:

Voir Chapitre 2.10.2 pour la correspondance des éléments logiques combinatoires. Les expressions %ECn contenant des appels de BFCs sont traduites par :

- Blocs opération : Opérateurs langage, les Fonctions ou de Comparaison,
- Blocs langage (FB)

Une expression contenant une succession de BFCs avec des "**liens nommés**" est traduite par une cascade de "**Bloc Operate**" dans un rung.

## 2.7-2 Paramétrage généraux

On désigne par paramètres généraux les informations suivantes :

- l'origine de la numérotation des Rungs dans le code PL7,
- la variable dite "de report" permettant de convertir les conditions de validation des Entités,
- la configuration cible,
- le type d'import des déclarations.

### Entités

Le paramétrage des entités consiste à la saisie :

- de l'origine de la numérotation des Rungs dans le code PL7
- de la variable de report pour la condition de validation

Convertisseur Orphée - PLATETH4 - %CC

**PARAMETRES GENERAUX**

Entités

**Déclarations**

VARIABLES

Bits internes

**Mots de 16 bits**

Mots de 32 bits

Réels

Bits de mots 16 bits

Bits de mots 32 bits

Bits de graphe

Tables de bits

**Tables de mots 16 bits**

Tables de mots 32 bits

Tables de réels

Chaînes de caractères

Constantes

E/S TOR et Numériques

**LOGIQUE COMBINATOIRE**

Nb max de contacts en série

Liens entre boîtes fonctionnelles

**FONCTIONS**

Evénements programmes

Boîtes fonctionnelles

**Numérotation**

Origine de la numérotation des Rungs

%L

Numéro de Rung

**Commentaire**

Sans étiquette Orphée

Si positionné, élimine les références aux %CC et % EC mises dans les commentaires issus d'Orphée

**Condition**

Variable de report associée aux conditions de validation des entités

Nombre de Bits 256

%M

Variable de report de type bit

**Paramètres**

Enregistrer Rappeler

Retourner Convertir

## Détermination de la Configuration cible

Le choix "Configuration cible" permet de paramétrer les plages des Variables à partir desquelles le Convertisseur effectuera les correspondances par défaut. Le paramétrage par défaut est :

- . %MX débutent à l'adresse 0,
- . %RX "à la suite des %MX",
- . %MW débutent à l'adresse 0,
- . %MD "à la suite des %MW",
- . %FD "à la suite des %MD",
- . %CH "à la suite des %FD".

Ce paramétrage est accessible en modification, uniquement avant l'ouverture d'une application. Par la suite, la Configuration cible est consultable à partir du menu "Services".

The screenshot shows a dialog box titled "Configuration cible". It contains the following sections:

- Bits:**
  - %MX / %MX(): Origine cible [0]
  - %RX / %RX():  A la suite des %MX,  Origine cible [ ]
- Mots 16 bits:**
  - %MW / %MW(): Origine cible [0]
- Mots 32 bits:**
  - %MD / %MD():  A la suite des %MW,  Origine cible [ ]
- Réels:**
  - %FD:  A la suite des %MD,  Origine cible [ ]
- Chaînes de caractères:**
  - %CH():  A la suite des %FD,  Origine cible [ ]

Buttons "OK" and "Annuler" are located on the right side of the dialog.

Dans Orphée, l'espace des variables est découpé en bloc, et un bloc est réservé à un type.

Exemple de répartition des noms constructeur des objets d' Orphée (A5000 sans extension RAM):	%MWN	de 0	à	4999
	%MDn	de 5000	à	5999
	%FDn	de 6000	à	6999
	%CHn	de 7000	à	11262

---

Sous PL7, l'utilisateur peut mélanger les types, en utilisant %MW0, %MD2 et %MW5.

Si le Convertisseur se contentait d'associer %MD5000 (Orphée) à %MD5000 (PL7), la plupart des repères associés dépasseraient les capacités de l'automate, obligeant l'utilisateur à ressaisir toutes ses variables. Aussi le Convertisseur utilise-t-il une configuration cible (par défaut ou saisie par l'utilisateur), qui précise pour chaque type s'il doit être mis **à la suite de** (choix par défaut) ou commencer à **Origine**..

Ainsi, si l'utilisateur souhaite que les %MD soient placés "à la suite des" %MW, le Convertisseur recherche le plus grand %MW rencontré (mot simple ou table) et fait commencer l'espace des %MD à l'adresse suivante. Si l'utilisateur veut réserver de l'espace pour d'autres déclarations, il peut alors saisir l'**Origine**: des %MD, c'est à dire l'indice du premier %MD généré. Si cette adresse n'est pas suffisamment grande, elle génère des conflits avec des mots ou tables de mots, et le Convertisseur présente à l'utilisateur une configuration conseillée (minimale) qu'il peut modifier avant de relancer l'analyse.

#### **Procédure conseillée:**

Il faut:

- choisir les valeurs par défaut,
- lancer la sélection et l'analyse,
- en cas de conflits, le Convertisseur propose une nouvelle configuration en fonction des variables à traduire.

1 2 3- Configuration de la Conversion 4

## Déclarations



Paramétrer la déclaration signifie choisir le mode d'import entre les 2 modes suivants :

### Mode prioritaire :

Si la variable est déjà déclarée dans l'application cible, elle sera écrasée par celle importée.

### Mode non prioritaire :

Le Convertisseur se limite à l'ajout de nouvelles variables

## 2.7-3 Paramétrage des variables

Le paramétrage est effectué par type de variable. Pour chaque type, il sera appelé :

- son contexte d'utilisation dans les Entités sélectionnées et dans l'application cible après conversion.
- le nombre configuré de variables de même type qui lui correspond dans l'application cible.



Exemple :

Variables Orphée                      Variables PL7

Utilisés	Proposés
Origine de %MX17 à %MX19 (5)	VIE_VAL %MX15 %M15
Cible de %M15 à %M19 (5)	VIE_FAZ %MX16 %M16
	VIE_TRAV %MX17 %M17
	VIE_ER1 %MX18 %M18
	VIE_ER2 %MX19 %M19

Visualisation des variables

Configuration de l'automate cible

Paramétrage par bloc

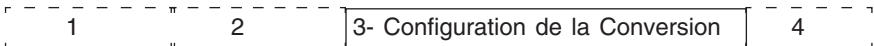
Deux méthodes sont données pour paramétrer les variables. Ces méthodes consistent à proposer :

1. une transcription, variable par variable. Elle est présentée avec la possibilité de modifier l'indice cible de la variable sélectionnée
2. un décalage par bloc de tout ou partie des variables vers un indice destination précisé par l'utilisateur (de .. à .. vers ...).

Pour la 1ère méthode, la procédure est la suivante :

- sélectionner une variable en défaut parmi celles qui sont visualisées,
- saisir l'indice cible dans la fenêtre de saisie prévue à cet effet.

La sélection multiple des variables proposées n'est pas possible.



Pour la 2ème méthode, le décalage peut se faire de 2 manières :

**1 - sans l'option pas de 1**

on décale l'indice de la 1ère donnée vers l'indice cible et ceux des autres variables de la même valeur de décalage tout en préservant les écarts d'origine.

**Exemple:**

Bloc à décaler: %MX10, %MX12, %MX13, %MX24 Vers 100

Bloc résultat: %MX100, %MX102, %MX103, %MX114

**2 - avec l'option pas de 1**

on décale l'indice de la 1ère variables vers l'indice cible . Pour les autres variables, on leur affecte incrémentalement le numéro suivant.

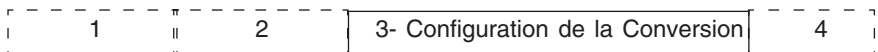
**Exemple:**

Bloc à décaler: %MX10, %MX12, %MX13, %MX24 Vers 100

Bloc résultat: %MX100, %MX101, %MX102, %MX103

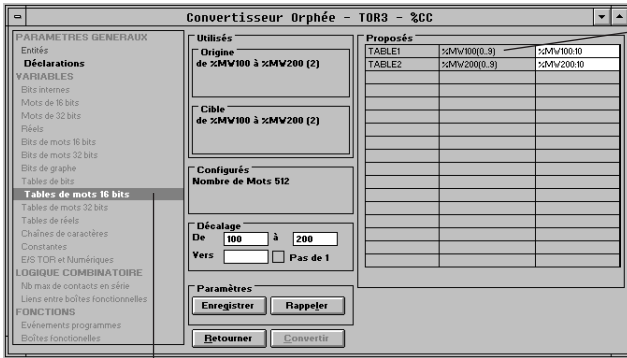
Note.:

- Quel que soit le type de modification de l'indice d'une variable (individuelle ou par bloc), **la double affectation est interdite y compris dans les tables.**
- Toute variable incluse dans le bloc est ré-affectée automatiquement même si son indice a préalablement fait l'objet d'une modification unitaire.



## Interface utilisateur, cas général

L'interface générale de paramétrage des variables est la suivante :

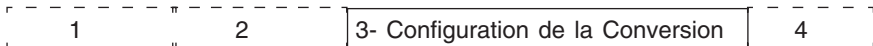


Affichage de la liste des variables du type choisi

Sélection de la rubrique nature de variables

Les types concernés sont les suivants:

- les bits internes,
- les mots de 16 bits,
- les mots de 32 bits,
- les réels,
- les bits de mots de 16 bits,
- les bits de mots de 32 bits,
- les tables de bits,
- les tables de mots de 16 bits,
- les tables de mots de 32 bits,
- les tables de réels,
- les chaînes de caractères,
- les constantes.



La visualisation et modification du paramétrage associé à une information de configuration est réalisée en sélectionnant la ligne correspondante (non "grisée"). Le paramétrage est alors affiché à la droite de la liste.

Dans le cas où une intervention de l'utilisateur est demandée, les paramètres "en défaut" sont signalés en Rouge .

Une information de configuration signalée en "grisé" et comprise entre "<>" signifie que des objets sont présents dans l'application d'origine mais le type n'est pas traduisible.

Rubrique avec des conflits en rouge

Variable "en défaut" nécessitant une intervention

Toute suppression du "défaut" s'accompagne de la mise à jour de la couleur en temps réel.

Le Convertisseur affiche "**INSUFFISANT**" en plus du nombre d'objets configurés si l'un des indices de variables proposées est supérieur au nombre de variables configurées.

On peut alors corriger de 3 manières différentes

1. en ré-affectant les variables dans les limites de la configuration,
2. en modifiant la configuration,
3. en "effaçant" (touches "Suppr" et "Entr") la ou les variables dépassant la configuration. Dans ce cas, le Convertisseur générera pour ces variables des symboles particuliers de la forme : ORPHEE\_"nom de la variable", qu'il faudra remplacer lors de l'import du code.

Comme tout autre "défaut", cette situation est bloquante et inhibe l'accès à la fonction "Convertir".

Rubrique en modification de variables avec conflits.

Variables à paramétrer

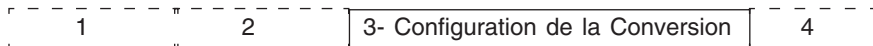
Paramétrage individuel de la variable.

### Procédure:

- sélectionner une rubrique en rouge,
- sélectionner chaque variable en rouge,
- saisir une valeur admissible dans la fenêtre de paramétrage individuel, ou .. effacer" la variable.

### Note:

Dans la phase de paramétrage de la conversion, toute modification de la configuration matérielle ou logicielle par l'utilisateur (utilisation simultanée du Convertisseur et de l'Editeur Configuration) se traduira dans le Convertisseur par la mise à jour automatique des paramètres de configuration et de l'affichage courant.



## Paramétrage des Bits de graphes

PL7 ne gère qu'un seul niveau de graphe. Le paramétrage des bits de graphes consiste à fournir deux variables de manoeuvre :

- une variable de type bit remplaçant tous les bits grafcet,
- une variable de type mot remplaçant tous les registres de type mot du grafcet (cf. 2.10.1).

### Procédure:

- sélectionner la rubrique **Bit de graphe**,
- saisir des valeurs pour les variables demandées. Ces valeurs doivent correspondre à des variables configurées dans l'automate cible.

L'interface est la suivante:

## Paramétrage des E/S et Numériques

Il s'effectue selon deux modes :

1. Variable par variable avec possibilité de modifier l'indice topologique.  
Le paramétrage est effectué dans le champ "Proposés". La fenêtre de saisie de l'indice est accessible en sélectionnant une variable dans la liste.
2. Par association des adresses topologiques des coupleurs dans les configurations matérielles.

Le champ "*Emplacements cartes*" présente les adresses des cartes d'E/S utilisées dans l'application Orphée. Pour chacune d'elle, le champ voie de départ précise l'indice de la première variable du coupleur. Le champ cible précise le numéro (adresse) du coupleur cible.

Le paramétrage est effectué dans le champ "*Emplacements cartes*" par association des emplacements des coupleurs suivi d'une proposition de correspondance de la voie de départ. Celle-ci peut être modifiée.

Il est possible de "compresser" les adresses cibles en sélectionnant l'option **Pas de 1**. Cette option a pour effet de proposer automatiquement des numéros de voies contiguës.

En cas de problème d'association (ex.: variable résultat en défaut configuration), les variables mal "associées" seront en rouge dans la liste.

Utilisation du 2ème mode  
par association d'adresses

Utilisation du 1er  
mode, avec saisie  
individuelle.

The screenshot shows the 'Convertisseur Orphée - TOR3 - %CC' window. On the left is a sidebar with categories like 'PARAMETRES GENERAUX', 'Déclarations', 'VARIABLES', 'E/S TOR et Numériques', etc. The main area is divided into 'Entrées' and 'Proposés'. Under 'Entrées', there is a section 'Emplacement Cartes' with fields for 'Origine' (SF 000), 'Cible', 'Voie de départ' (20), and a checkbox 'Pas de 1'. Below this are 'Paramètres' buttons: 'Enregistrer', 'Rappeler', 'Retourner', and 'Convertir'. The 'Proposés' section contains a table with columns 'ES\_IN2' and '2dW20'. A 'Saisie 2dW' field is at the bottom right.

Voie de départ

Numéro du coupleur

**Note :** Les champs cible et voie de départ doivent être cohérents avec la configuration de l'automate cible.

1 " 2 - 3- Configuration de la Conversion - 4

### 2.7-4 Limitations et restrictions

Lors de l'activation de la traduction, les expressions sont classées en trois catégories:

- celles qui ne sont pas traduisibles,
- celles qui nécessitent un paramétrage,
- celles qui sont traduisibles.

Compatibilité avec les processeurs V 1.5 :

Certaines instructions (index immédiat, ROL et ROR quand le nombre de bits de décalage est une variable %MW, PULSOR) ne seront pas acceptées au moment de l'import, lorsque on utilise, dans la configuration matérielle, un processeur de version 1.5.

#### Les conditions qui nécessitent un paramétrage sont :

Expression de 11 contacts et une bobine	Lors de la conversion d'une %EC comportant 11 contacts et une bobine, le Convertisseur la découpe en 2 rungs avec une variable "report".
Enchaînement de BFCs	Lors d'une conversion d'une expression comportant un enchaînement de BFCs "convertibles" avec un lien binaire DIRECT (NON renseigné), le Convertisseur demande la saisie d'un bit de travail. Le nombre maximum de variable report est calculé sur l'expression la plus "complexe".
BFC temporisateur et compteur/decompteur	saisie d'un mot de travail simple (pour BFC TON) ou double (pour BFC PULSOR) ou d'un numéro d'instance (cf § 2.7.6.1)



---

## AVERTISSEMENT

### COMPORTEMENT INATTENDU DE L'APPLICATION

La conversion de la fonction PULSOR met le paramètre Enable de cette fonction à la valeur TRUE. De ce fait même si la fonction n'est plus exécutée, la sortie reste dans l'état.

Prenez en compte les instructions suivantes pour avoir le fonctionnement correct de la fonction PULSOR.

**Le non-respect de cette directive peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.**

Lors d'une conversion d'une BFC PUSLOR, le convertisseur crée deux rungs.

Le premier rung avec un timer dont la sortie actionne un bit (%Mi).

Dans le deuxième rung, le bit (%Mi) valide l'appel d'un bloc OPERATE dans lequel se trouve la fonction PULSOR. Le paramètre Enable de cette fonction est bloqué à la valeur TRUE. Quand la sortie du timer invalide le bit, la fonction PULSOR ne s'exécute plus car le bloc operate n'est plus appelé, mais la sortie de la fonction reste dans l'état où elle se trouve à ce moment.

Pour corriger cette mauvaise conversion, mettez le bit %Mi (activé par le timer) dans le paramètre Enable de la fonction PULSOR et validez en permanence le bloc operate.

### Les conditions qui interdisent la traduction sont :

Expressions logiques

- Expressions contenant un %CONT,
- Expressions contenant au moins une initialisation des registres Graphe: MIN, MAX et NUM (%ACQGRAF),

Utilisation de BFCs et BFUs

- Expressions contenant des BFUS, Expressions contenant des BFCs non convertibles (cf 2.7.6.2 la liste des BFCs convertibles), Liaison BFC/BFC avec des liens binaires indirects.

## 2.7-5 Paramétrage de la logique combinatoire

### Règle générale:

Chaque %EC est traduite en un rung.

La transcription fonctionnelle d'une %EC :

- soit réalisée complètement.
- soit un **rung vide**, en cas d'impossibilité.

La transcription d'une %EC complète peut nécessiter un paramétrage particulier.

La liste des cas d'impossibilités est donnée au paragraphe "Limitations et restrictions" (cf 2.7.4.).

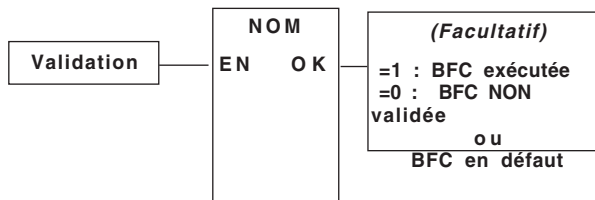
## 2.7-6 Paramétrage des boîtes fonctionnelles

**Rappel** : Dans l'Atelier Orphée, les BFCs sont classées par familles. Le Convertisseur Ladder ne prend en compte que les familles associées au langage Série 1000 de base. Les familles suivantes ne seront donc pas prises en compte:

- gestion de graphes,
- mesures et régulation,
- régulation,
- axe,
- communication,
- sûreté.

### Règles générales pour la conversion des BFCs :

D'une manière générale, l'environnement d'exécution d'une BFC Orphée se présente comme suit:



Les règles de la conversion sont les suivantes :

**Règle 1 :**

SI la sortie OK de la BFC n'est pas programmée

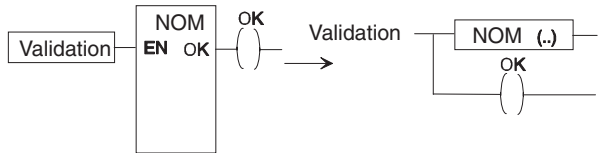
la conversion sera réalisée de manière bijective.



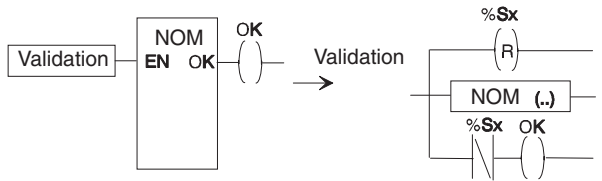
**Règle 2:**

SI la sortie OK de la BFC est programmée

1er Cas : l'opération réussie systématiquement.



2ème Cas : l'opération peut échouer



**Cas particuliers:**

**Les temporisateurs**

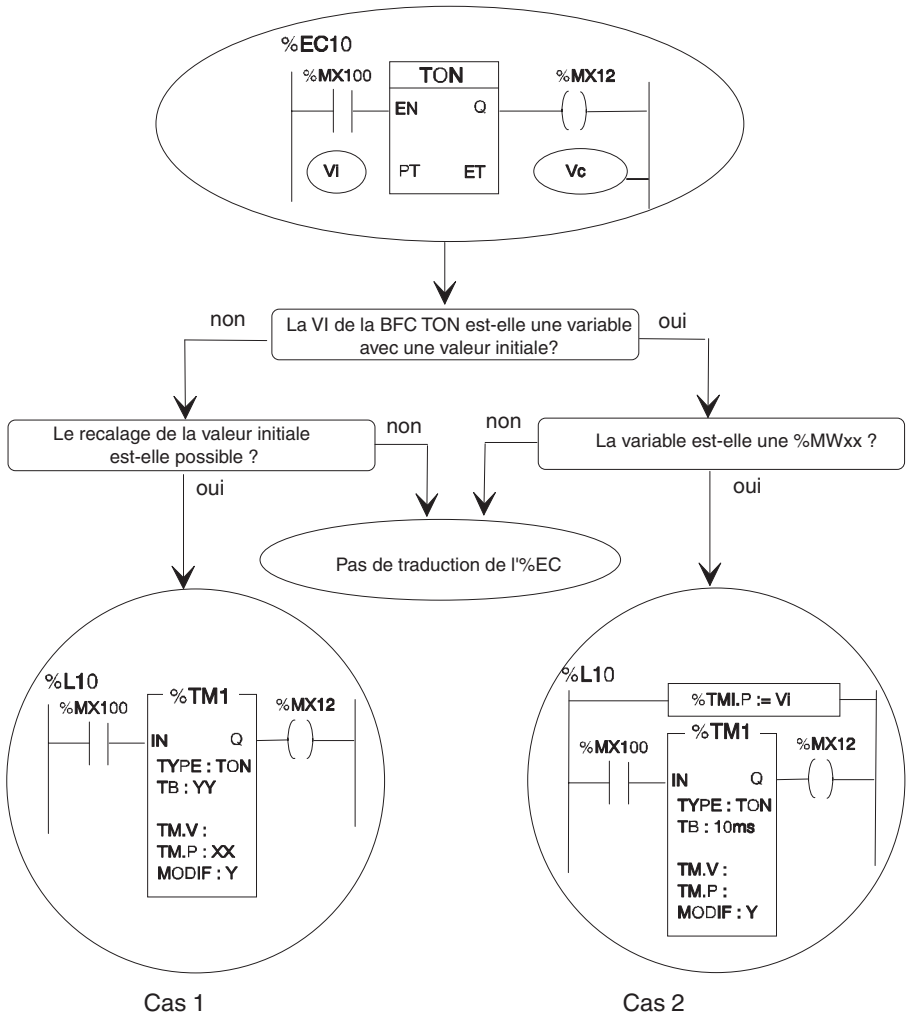
Ce cas concerne le paramétrage de la BFC : TON.

Les fonctions équivalentes dans le langage cible PL7 sont réalisées à partir du Bloc Fonction %TMI pouvant être configuré en TON, avec un numéro d'instance "i" (de 0 à un nombre configuré) à préciser lors de l'association.

Dans les BFC Orphee, le couple donnant la limite de la temporisation : Valeur initiale/ base de temps est à réajuster en fonction de la limite "configurable" du %TMI : base temps (Cste) = 1mn, 1s, 100ms au 10ms/ Valeur initiale (Cste) = {0..99991}.

Ainsi une BFC TON ayant 45000 comme valeur initiale sera transformée en TIMER (TON, 100ms, 4500).

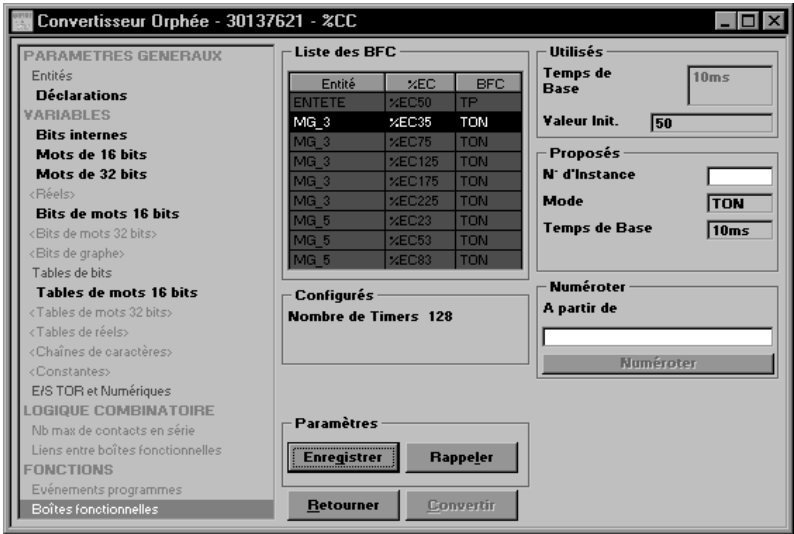
L'algorithme de traduction est décrit avec l'exemple de BFC TON suivant :



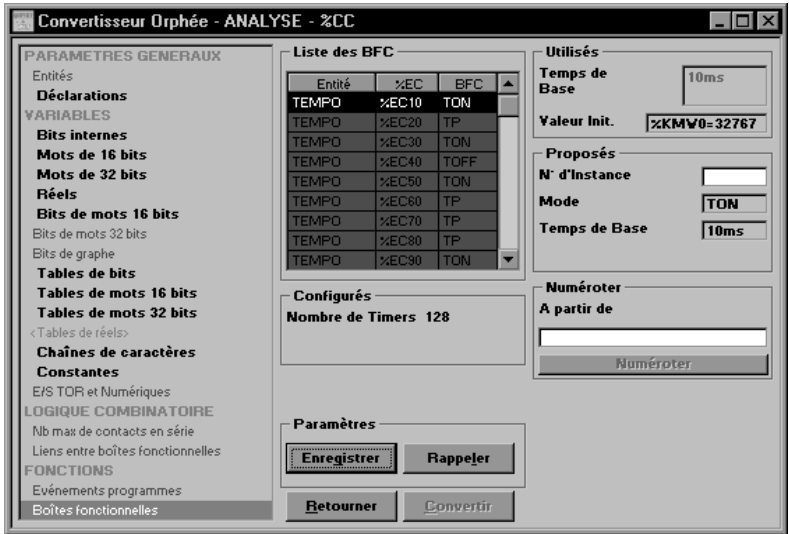
**Remarque:**

La variable Vc de la BFC contenant la valeur courante est ignorée.

Cas 1 : Pas de valeur initiale



Cas 2 : Avec valeur initiale



## Le compteur/décompteur

La traduction de la BFC comptage/décomptage COUNT en une fonction équivalente dans le langage cible PL7 est réalisée à partir d'une "OF spécifique Série 1000", identique à la BFC COUNT.

La mémorisation des états CU et CD, ainsi que le "report" des entrées de type <expression bit> sont réalisés à l'aide d'une variable numérique passée en paramètre à l'OF.

Cette variable est saisie lors du paramétrage des "Boîtes fonctionnelles" suivante:

Convertisseur Orphée - TOR3 - %CC

**PARAMETRES GENERAUX**

Entités

**Déclarations**

**VARIABLES**

Bits internes

Mots de 16 bits

Mots de 32 bits

Réels

Bits de mots 16 bits

Bits de mots 32 bits

Bits de graphe

Tables de bits

**Tables de mots 16 bits**

Tables de mots 32 bits

<Tables de réels>

Chaînes de caractères

Constantes

E/S TOR et Numériques

**LOGIQUE COMBINATOIRE**

Nb max de contacts en série

Liens entre boîtes fonctionnelles

**FUNCTIONS**

Evénements programmes

Boîtes fonctionnelles

**Liste des BFC**

Entité	%EC	BFC
VELOCI	%EC40	TON
VELOCI	%EC70	COUNT
C_INIT	%EC10	TON

**Proposés**

Variable de mémo.

%MW

**Configurés**

**Nombre de Mots 512**

**Paramètres**

**Enregistrer** **Rappeler**

**Retourner** **Convertir**

1

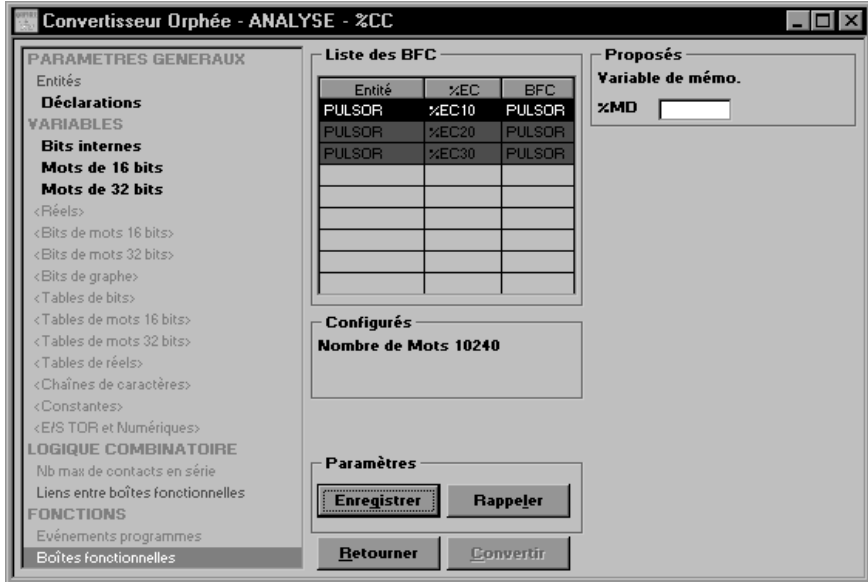
2

3- Configuration de la Conversion

4

## Pulsor

Pour la BFC PULSOR, la mémorisation des états internes pour l'équivalent en langage PL7 est réalisé à l'aide d'une variable %MD.



1

2

3- Configuration de la Conversion

4

## Liste des BFCs acceptées par le Convertisseur

FAMILLE	BFC	DESIGNATION
CALCUL MATH.	<i>ADD</i>	addition de valeurs
	<i>SUB</i>	soustraction de valeurs
	<i>MUL</i>	multiplication de valeurs
	<i>DIV</i>	division de valeurs avec reste
	<i>SQRT</i>	racine carrée avec reste
OPERATIONS LOGIQUES	<i>AND</i>	ET logique
	<i>NEG</i>	négation (complément) logique
	<i>OR</i>	OU logique
	<i>XOR</i>	OU exclusif
DECALAGES	<i>ROL</i>	circulaire à gauche de n bits
	<i>ROR</i>	circulaire à droite de n bits
	<i>SHL</i>	à gauche de n bits avec remplacement à 0 et récupération. des bits décalés
	<i>SHR</i>	à droite de n bits avec extension de signe et récupération des bits décalés
	<i>SHRZ</i>	Décalage à droite de n bits avec remplacement à 0 et récup. des bits décalés
TEMPOS/ COMPTEUR	<i>COUNT</i>	compteur et décompteur avec dépassement de seuil min et max
	<i>PULSOR</i>	Sortie créneau (Remarque sur la conversion : La sortie ET décroît sous ORPHEE et croît sous PL7. La sortie ET est un mot double sous ORPHEE et un mot simple sous PL7). Voir avertissement page 2/26.
	<i>TON</i>	temporisation à la montée
COPIE/ CONVERSION	<i>COPY</i>	recopie d'une variable dans une autre
	<i>ASCII_M</i>	conversion de ASCII en entier
	<i>BIN_DCB</i>	conversion de décimal en DCB
	<i>DCB-BIN</i>	conversion de DCB en binaire
	<i>FD_M</i>	conversion de flottant en entier
	<i>GRAY_BIN</i>	conversion de Gray en binaire
	<i>M_FD</i>	conversion d'entier en flottant
	<i>MD_MW</i>	conversion entier 32bits en entier 16 bits
	<i>MW_MD</i>	conversion entier 16 bits en entier 32 bits
<i>M_ASCII</i>	conversion d'entier en ASCII	



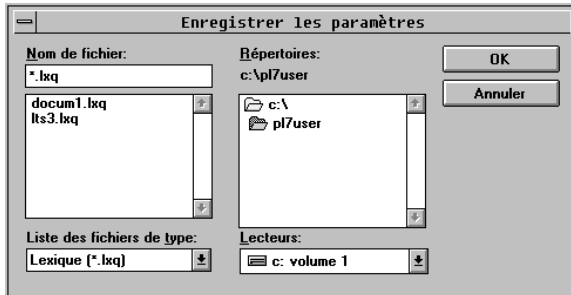
FAMILLE	BFC	DESIGNATION
COPIE/ de CONVERSION (suite)  table	<i>WD-BIT</i>	Transfert d'un mot ou table de mots dans une table de bits (* Table mots dans table de bits avec désignation des rangs Origine et Destination, * Table Mots double dans table de bits avec désignation des rangs Origine et Destination)
	<i>BIT_WD</i>	Transfert d'une table de bits dans un mot ou une de mots * Table de bits dans table Mots avec désignation (des rangs Origine et Destination, * Table de bits dans table de Mots double avec désignation des rangs)
TESTS	<i>COMP</i>	comparaison de 2 valeurs
	<i>COMP4</i>	comparaison d'une valeur à 4 autres
TRAITEMENT CHAINES	<i>CONCAT</i>	concaténation de 2, 3 ou 4 chaînes de caractères
	<i>LENGTH</i>	calcul de la longueur effective d'une chaîne
	<i>MID</i>	extraction d'une sous-chaîne
	<i>COMPCH</i>	comparaison de chaînes
	<i>DELETE</i>	Effacement d'une sous-chaîne
	<i>INSERT</i>	Insertion d'une sous-chaîne
	<i>REPLACE</i>	Remplacement d'une sous-chaîne
TRAITEMENT CHAINES	<i>FIND_EQ</i>	Recherche du premier élément égal à une valeur
	<i>OCCUR</i>	Calcul de l'occurrence d'une valeur

## 2.8 Mémorisation et Rappel des paramètres

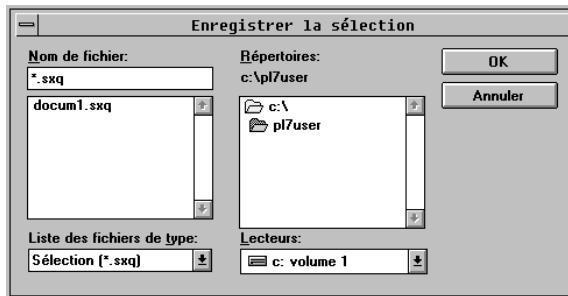
La fonction "Enregistrer" permet de sauvegarder dans un fichier l'ensemble des paramètres (modifiés ou pas) de chaque rubrique.

L'extension de ces fichiers sont:

- \*.lxq pour les paramètres de conversion,



- \*.sxq pour la sélection des variables.



Le paramétrage reste mémorisé tant que le Convertisseur est actif. Entre 2 sessions de conversion, la mise à jour des paramètres est effectuée suivant le principe de la fonction **Rappeler** décrite ci-dessous.

La fonction **Rappeler** permet d'associer aux informations de configuration en cours les paramètres de chaque rubrique préalablement enregistrés dans un fichier.

---

Ne sont pas sauvegardés:

- les associations des BFCs,
- le type d'import,
- le numéro du premier rung.

La configuration cible n'est pas enregistrée, il appartient à l'utilisateur de la noter avant de quitter pour la ré-utiliser.

### Note

Les fichiers de paramètres sont liés à une conversion. On protège donc les fichiers de sauvegarde avec les noms des applications Orphée et PL7. Toute tentative d'utilisation de ce fichier sur d'autres applications entraîne l'affichage de messages d'erreurs et d'avertissement

### Exemple:

- 1er cas : si le fichier \*.lxq possède la correspondance %MX0<=>%M10 et si le paramétrage en cours donne %MX0 <=> %M0 alors le service "Rappeler" réalise la correspondance finale %MX0<=>%M10.
- 2ème cas : si le fichier \*.lxq possède la correspondance %MX20 <=> %M30 et si %MX20 n'intervient pas dans le paramétrage en cours alors le service "Rappeler" est sans effet sur le paramétrage en cours.

## 2.9 Conversion

La fonction "Convertir" permet de lancer la conversion proprement dite, si toutes les informations de configuration à caractère obligatoire sont renseignées.

La conversion terminée, le rapport s'affiche automatiquement.

Le rapport est découpé en 4 parties, regroupant les données d'environnement, les conflits à résoudre, le récapitulatif de la conversion et la liste des variables utilisées. Chaque partie est décrite dans un des chapitres suivants.

### 2.9-1 Le bloc de contexte

```

*****
                                CONTEXTE
*****
Fichier généré :ANALYSE.LD
Date de conversion : le 05/11/1997
De l'application Orphée ANALYSE vers l'application PL7 <Sans nom>
Convertisseur Code Orphée V 3.0

```

On y retrouve le fichier généré, la date de conversion, le nom des applications Orphée (source) et PL7 (destination), et la version du Convertisseur..

### 2.9-2 Les éléments à configurer

```

*****
                ELEMENTS A CONFIGURER AVANT IMPORT
*****
PL7           Symbole      Orphée
%M1025         TOTALRX      %RX0
%M1026         PAsPAR       %RX1
Entité      Expression    Mode PL7
%TM512         TEMPO        EC10      TON
SR PL7 (présence obligatoire à l'import)
SR0

```

Trois catégories:

- les variables dépassant les capacités actuelles de l'automate (Hors Conf)
- les Timer ou mots de manoeuvre des BFC qui sont Hors Conf
- les Sub-Routine (SR) qui sont utilisées dans le code, et qui doivent être déclarées dans la tâche avant import.

1	2	3	4 - Conversion
---	---	---	----------------

### 2.9-3 Le résultat de conversion du code

```

*****
                                RESULTAT DE LA CONVERSION
*****
Entite      Nom      N° Rung

%CC0        TESTBIT  %L0
  %EC10      Traduite
  %EC20      Traduite A compléter à l'import.
  %EC30      Non traduite BFC COPY : paramètres
                                ... refusés.
  %EC40      Non traduite BFC ASCII_F intraduisiblee
%CC45       CFBTOTAL %L1      Entité avec Condition de
validation
  %EC50      Non traduite Les liens non directs ne sont
                                pas traduits.
                                L'expression est abandonnée.
  %EC60      %L2     Traduite
  %EC70      Non traduite L'expression est un OU de 11
                                contacts. Pas de traduction
  %EC80      Non traduite Variable inconnue!
  %EC90      Non traduite Pas assez de lignes pour
                                l'expression.
                                Pas de traduction.
  %EC100     Non traduite La variable suivante n'a pu
                                être identifiée EPNOMME

Saut de Validation %L3

```

Toute entité génère au moins une ligne rappelant son numéro, son nom et l'étiquette associée (obligatoire)

Toute expression (%EC) génère une ligne signalant la réussite ou l'échec de la conversion.

Pour les entités, le seul message possible est "**Entité avec Condition de validation**": l'entité a une condition de validation, celle-ci est générée avant la première expression.

Pour les expressions (%EC), on distingue 2 niveaux de message:

**"Traduite"**: l'expression est générée, avec peut-être une petite limitation :  
 "A compléter à l'import": une des variables n'a pas de traduction, il faudra compléter le code lors de l'import

**"Non Traduite"**: l'expression n'est pas générée, on ne trouvera dans le code qu'un commentaire et la raison de l'échec.

Les raisons de la "non traduction" sont les suivantes:

"BFC xxx intraduisible"	la BFC citée n'a pas d'équivalent en PL7
"BFC xxx: paramètres refusés"	la BFC existe en PL7, mais n'accepte pas ces types de paramètres
"Les liens non directs ne sont pas traduits"	un enchaînement de BFC contient une liaison binaire complexe, qui n'est pas convertible.
"L'expression est un OU de 11 contacts" un	l'expression compte 2 lignes occupant 11 contacts chacune, la traduction de cette expression engendre un réseau avec plus de 10 contacts en série; le OU ne peut être traduit.
"Pas assez de ligne PL7 pour l'expression"	l'expression équivalente occupe plus de 7 lignes en PL7
"Variable inconnue! "	l'expression contient un paramètre non renseigné, intraduisible
"La variable suivante n'a pu être identifiée" faire	en général, le code contient un appel à un Événement Programmé symbolisé, et le Convertisseur ne peut faire le lien entre Symbole et numéro d'EP

Une ligne supplémentaire peut être trouvée. Exemple: "**Saut de Validation %L5**". Elle indique que la dernière entité générée comportait une condition de validation, qui nécessite une étiquette de saut. On a donc généré un rung supplémentaire (%L5) pour obtenir cette étiquette.

1	2	3	4 - Conversion
---	---	---	----------------

## 2.9-4 le récapitulatif des données

=====			
VARIABLES ET BFC			
Symbole	Orphée	PL7	Défaut
TOTALMX	%MX0	%M0	
PASCTMX	%MX1	%M1	
TOTALMW	%MW0	%MW0	(Supp. de Tempo)
PASCMW	%MW1	%MW1	(Supp. de Tempo)
BITMD	%MD4002:X16	ORPHEE_MD4002X16	<b>A Renseigner</b>
LONGTRX	%RX330(0..50)	%M1355:51	<b>Symbole ignoré</b>
TOTALCH	%CH8000(0..9)	%MB1846:10	<b>V Init. ignorée</b>
<Instance des BFC>			
Entite	Expression	Mode	PL7
TEMPO	EC10	TON	%TM512
COUNT	EC30	COUNT	%MW78

Cette partie donne l'ensemble des associations utilisées pour convertir le code. On y trouve donc les équivalences entre variables Orphée et PL7 mais aussi les affectations de timer et de mots de manoeuvre des BFC

### Les défauts possibles sont:

" <b>A Renseigner</b> "	la variable n'est pas associée à un repère PL7, il faudra compléter l'outil rung où apparaît cette variable est traduit).
" <b>Conflit de symbole</b> "	le repère possède un autre symbole dans l'application ou dans l'ensemble des variables traduites.
" <b>Symbole ignoré</b> "	le symbole rentre en conflit avec une autre déclaration, il n'est pas importé ou générera un message.
" <b>Mot réservé PL7</b> "	le symbole est un mot réservé PL7
" <b>V Init. ignorée</b> "	pour les chaînes, on rappelle qu'il n'y a pas de valeurs initiales dans PL7.
" <b>(Supp. de Tempo)</b> "	le mot ou double mot était utilisé comme valeur actuelle d'une BFC Timer. La conversion ne générant pas le code permettant de récupérer cette valeur, le mot n'est plus lié à la BFC.

1	2	3	4 - Conversion
---	---	---	----------------

## 2.10 Tableau de correspondance

### 2.10-1 Tableau de correspondance des variables

Données ORPHEE	Données PL7	Conversion	Symbole
%MXn	%Mn ou %MXn (1)	AUTOMATIQUE	RECUPERABLE
%RXn	%Mn ou %MXn	AUTOMATIQUE	RECUPERABLE
%MWn	%MWn	AUTOMATIQUE	RECUPERABLE
%MWn:Xi	%MWn:Xi	AUTOMATIQUE	RECUPERABLE
%MDn	%MDn	AUTOMATIQUE	RECUPERABLE
%MDn:Xi	%MWin:Xi %MXn	ASSISTEE	RECUPERABLE
%FDn	%MFn	AUTOMATIQUE	RECUPERABLE
%MXn(0..k)	%Mn:L	AUTOMATIQUE	RECUPERABLE
%MXn(%MWj) %MXn(i)	%Mn(%MWj] %M(n+i)	AUTOMATIQUE	RECUPERABLES
%RXn(0..k)	%Mn:L	AUTOMATIQUE	RECUPERABLE
%RXn(%MWj) %RXn(i)	%Mn[%MWj] %M(n+i)	AUTOMATIQUE	RECUPERABLES
%MWn(0..k)	%MWn:L	AUTOMATIQUE	RECUPERABLE
%MWn(%MWj) %MWn(i)	%MWn[%MWj] %MWn(i)	AUTOMATIQUE	RECUPERABLES

#### (1) - ATTENTION:

Les bits monostables n'ont pas d'équivalent en PL7.  
TOUS LES BITS SONT MEMORISES.

1	2	3	4 - Conversion
---	---	---	----------------



Données ORPHEE	Données PL7	Conversion	Symbole
%MDn(0..k)	%MDn:L	AUTOMATIQUE	RECUPERABLE
%MDn(%MWj) %MDn(i)	%MDn[%MWj] %MDn(i)	AUTOMATIQUE	RECUPERABLES
%FDn(0..k)	%MFn:L	NON TRADUITE	NON RECUPERABLE
%FDn(%MWj) %FDn(i)	%MFn[%MWj] %MFn(i)	NON RECUPERABLE	NON RECUPERABLE
%CHn(0..k)	%MBn:L	AUTOMATIQUE	RECUPERABLE
%KMWn	%KWn	AUTOMATIQUE	RECUPERABLE
%KMDn	%KDn	AUTOMATIQUE	RECUPERABLE
%KHWn	non	AUTOMATIQUE	RECUPERABLE
%KHDn	non	AUTOMATIQUE	RECUPERABLE
%KFDn	%KFn	AUTOMATIQUE	RECUPERABLE
%KCHn(0..k)	%KBn:L	AUTOMATIQUE	RECUPERABLE
%IXn	%In.MOD.vole	ASSISTEE	RECUPERABLE
%QXn	%Qn.MOD.voie	ASSISTEE	RECUPERABLE
%IWn	%IWn.MOD.vole	ASSISTEE	RECUPERABLE
%IDn	%IDn.MOD.vole	ASSISTEE	RECUPERABLE
%QWn	%QWn.MOD.voie	ASSISTEE	RECUPERABLE
%QDn	%QDn.MOD.vole	ASSISTEE	RECUPERABLE

1	2	3	4 - Conversion
---	---	---	----------------

---

Données ORPHEE	Données PL7	Conversion	Symbole
%GMm:Xn RECUPERABLE		ASSISTEE	NON
%GEm:Xn RECUPERABLE		ASSISTEE	NON
%XEm:Xn RECUPERABLE		ASSISTEE	NON

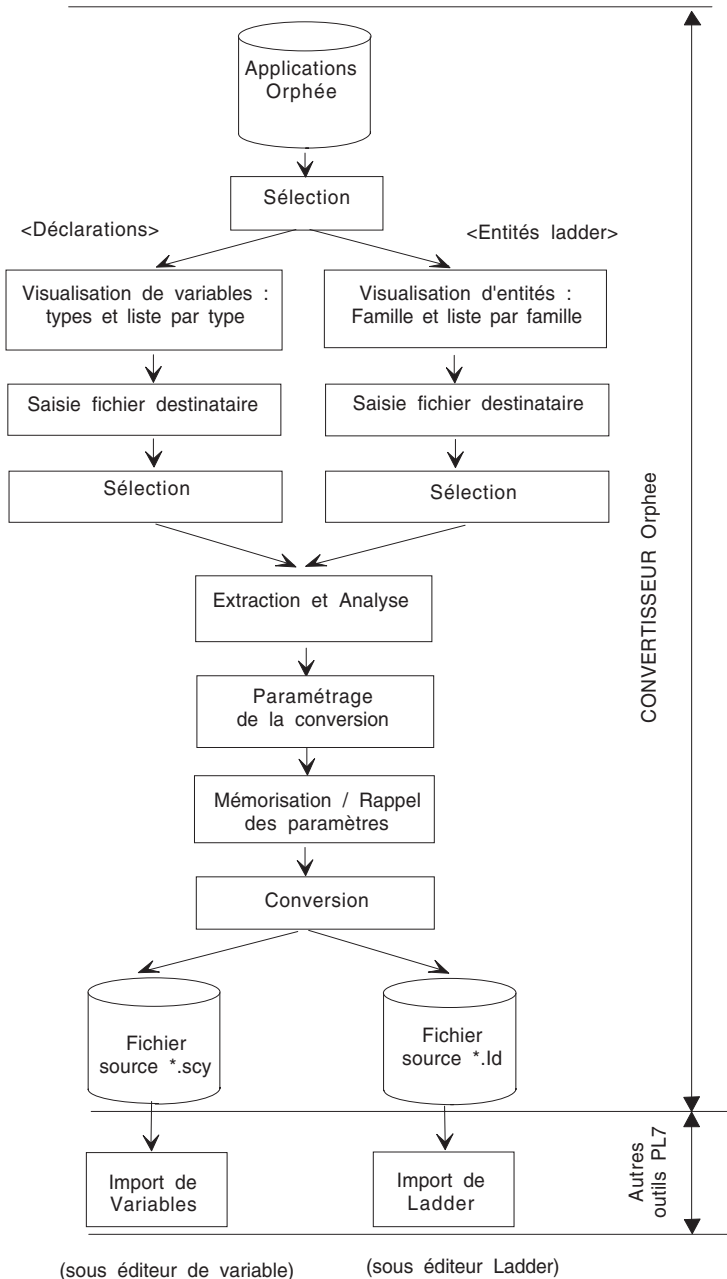
---

1	2	3	4 - Conversion
---	---	---	----------------

## 2.10-2 Tableau de correspondance de la logique combinatoire traduisible

Données ORPHEE		Données PL7	
Contact ouvert		Contact ouvert	
Contact fermé		Contact fermé	
Contact front montant		Contact front montant	
Contact front descendant		Contact front descendant	
Bobine ouverte		Bobine ouverte	
Bobine fermée		Bobine fermée	
Bobine SET		Bobine SET	
Bobine RESET		Bobine RESET	
Saut à une %ECn:	$\longrightarrow$ %ECn	Saut à un RUNG	$\longrightarrow$ %Li
Arrêt du prog.	$\longrightarrow$ %STOP	Bobine halt :	$\leftarrow$ HALT $\rightarrow$
Init du prog.	$\longrightarrow$ %INIT	Par mise à 1 du bit %S0	
Début masquage des événements	$\longrightarrow$ %MSKEVT	instruction MSKEVT dans bloc operate	
Fin masquage des événements	$\longrightarrow$ %DMSKEVT	instruction UNMASKEVT	
Appel élément programme		SR non paramétré	

3.1 Synoptique



(sous éditeur de variable)

(sous éditeur Ladder)

---

## 3.2 Préparation préliminaire

---

Tout d'abord, il faut placer les applications à récupérer dans le SERVROOT du répertoire d'exécution de PL7 Junior, ou dans un SERVROOT déporté. Se reporter au § 2.2 pour la récupération des applications.

Il est ensuite conseillé de disposer du dossier Orphée de l'application (données, références croisées et programme).

---

## 3.3 Récupération des variables uniquement

---

On souhaite récupérer les déclarations de l'application Orphée, c'est à dire les symboles et les commentaires. Les valeurs initiales ne sont pas récupérées, sauf pour les constantes.

---

### 3.3-1 Sur une nouvelle application PL7

L'espace de variables de l'application ne contient que les variables importées. On n'a donc pas de problèmes avec l'existant PL7 au moment de la saisie des associations, ni au moment de l'import. Il ne reste que les conflits entre les variables importées et les erreurs de configuration.

#### Exemple d'utilisation

1. A partir du dossier de l'application Orphée, préparer la **liste des E/S** et des déclarations à récupérer.
2. Ouvrir PL7 et créer une application. Constituer la configuration des **modules d'entrées/sorties** selon la liste précédente, et configurer le nombre de bits, mots, timers et constantes nécessaires .  
Ceci permet d'éviter de trop fréquents aller/retour entre l'éditeur de configuration et le convertisseur  
Note. Cette opération peut n'être faite qu'avant l'import, à l'aide du rapport de conversion.
3. Après validation des configurations, enregistrer l'application.  
On possède ainsi une base stable pour les enregistrements de paramètres.
4. Lancer le convertisseur, choisir l'application Orphée et l'ouvrir.
5. Choisir la rubrique **Déclaration**, la liste des types de variables se constitue.
6. Choisir un type de variable puis les variables dans le type, réitérer sur les autres types.
7. Demander **Analyser**
8. Vérifier ou saisir les associations et résoudre les cas de conflits présentés en rouge.

9. Demander à “Convertir” les déclarations.
10. Lire attentivement le rapport de conversion.
11. Ouvrir l’éditeur de Variables, aller dans le menu **Fichier**, choisir **Importer**.
12. Enregistrer l’application PL7 obtenue, fermer le convertisseur.

---

### 3.3-2 Sur une application PL7 existante

Le principal problème est la gestion des conflits entre les variables ajoutées et les variables existantes. La démarche est la même, hormis le fait que la saisie d’une configuration cible peut être nécessaire. On doit également surveiller très attentivement les conflits de symboles pour ne pas perdre des variables. Enfin, l’action d’import dans l’éditeur de variables doit être fait en mode **Dialogue** pour résoudre les conflits. Il est conseillé de disposer du dossier PL7 de l’application cible avant de commencer.

---

## 3.4 Récupération des fonctions d'Automatismes et de logique combinatoire (entités + variables)

---

Le but est de récupérer des fonctions de l'application Orphée dans une nouvelle application PL7. Comme l'import du code seul n'a aucun sens, le fichier généré contient les déclarations nécessaires à ce code. Les conflits peuvent donc naître d'un mauvais paramétrage de la configuration cible ou de mauvaises associations de variables. Le seul conflit généré par le code en lui-même est une mauvaise numérotation des rungs.

---

### 3.4-1 Sur une application PL7 vierge

Pour la nouvelle application PL7, la configuration cible se base uniquement sur les besoins des fonctions traduites.

Il est conseillé d'utiliser des numéros de rung croissant, en s'aidant du rapport de conversion précédente.

#### Exemple d'utilisation

1. A l'aide du dossier de l'application Orphée, lister les variables nécessaires à la conversion et noter surtout les indices maximum pour les bits (mémorisés, et non mémorisés), les mots, les double-mots et les réels. De même lister les besoins en Entrées/Sorties.
2. Ouvrir PL7, créer une application. A l'aide de la liste précédente, configurer les modules d'entrées/sorties et la mémoire de l'automate. Valider et enregistrer l'application.
3. Lancer le convertisseur. Saisir la configuration cible.
4. Choisir et ouvrir l'application Orphée,
5. Choisir les types d'entités puis les entités à convertir. la sélection peut être contiguë ou non.
6. **ATTENTION:** la sélection sera convertie en un seul fichier résultat, qui sera importé dans un seul module programme. Exemple: plusieurs EP ne pourront être importés que dans une seule SR.
7. Demander à **Analyser** la sélection, le code et ses données sont extraits.
8. Vérifier ou saisir les paramètres de la conversion.
9. Demander à **Convertir** le code. Lire attentivement le rapport de conversion.
10. Ouvrir l'éditeur Ladder dans une section, SR,.... Aller dans le menu fichier, option Importer.
11. Enregistrer l'application PL7, fermer le convertisseur.

---

### 3.4-2 Sur une application PL7 existante

Les principales différences sont:

- la détermination de la configuration cible: il faut prendre en compte les besoins de l'application Orphée et ceux de l'application PL7.
- le choix du numéro du premier rung généré qui ne doit pas créer de conflits avec l'existant.

Il faut ensuite être très attentif aux conflits de symboles pour tous les types de variables, et de choisir une conversion en **MODE NON PRIORITAIRE**.

---

## 3.5 Récupération du maximum d'une application

Nous conseillons d'effectuer dans l'ordre les actions de conversion des déclarations puis celle du code. Il est conseillé d'importer le fichier de variables avant de commencer la conversion de code.



