



Ce document a pour but de décrire la configuration d'un projet gérant la redondance sous tous ses aspects : gestion de la redondance dans le projet en lui-même, redondance entre 2 automates, redondance entre 2 bases de données, redondance côté clients légers. Ce document s'accompagne donc du projet « SystemRedundancy » qui servira de supports pour le reste des explications.

Dans ce projet la redondance est gérée ainsi :

- le projet est identique pour les 2 runtimes et les 2 runtimes sont lancés.
- Un seul des 2 (défini comme serveur primaire) est actif et communique réellement avec les automates, la base de données et les clients légers.
- L'autre runtime est exécuté mais toutes les fonctionnalités sont désactivées et seront activées dans le cas où le runtime primaire ne répond plus. Le runtime secondaire prendra alors la main et les clients légers basculeront automatiquement dessus.

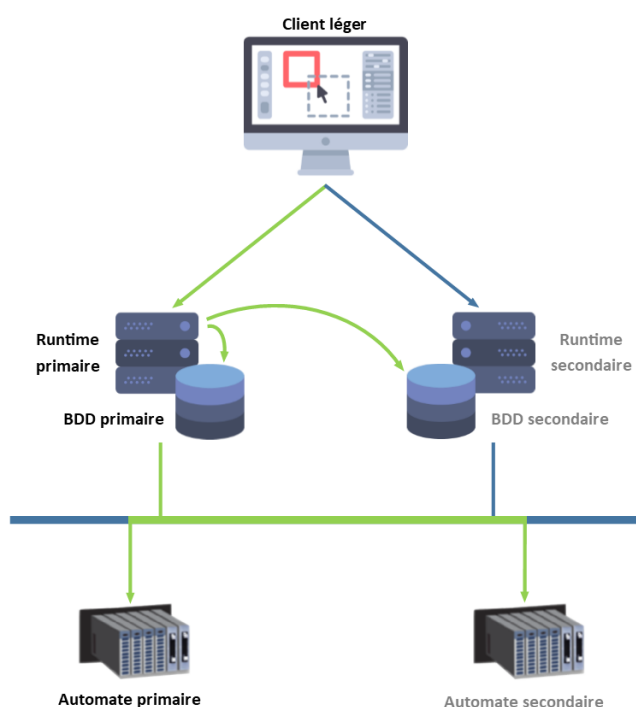


Figure 1 – Schéma redondance : état 1

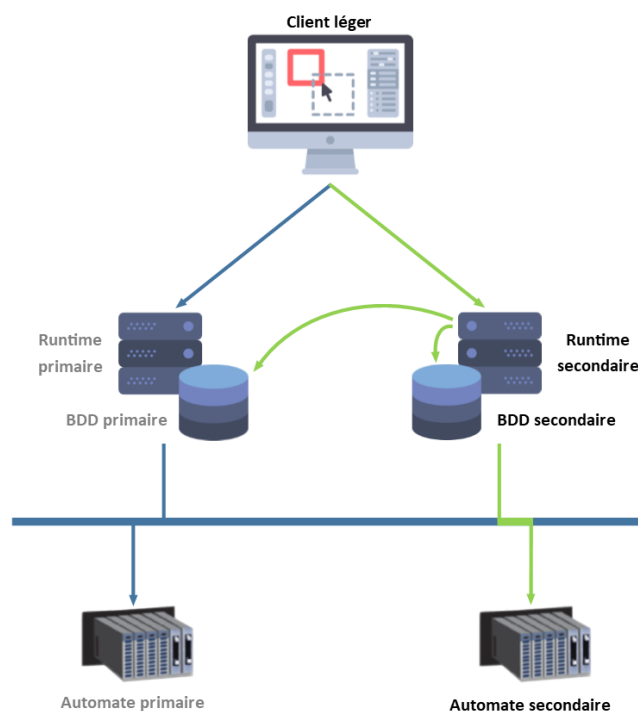
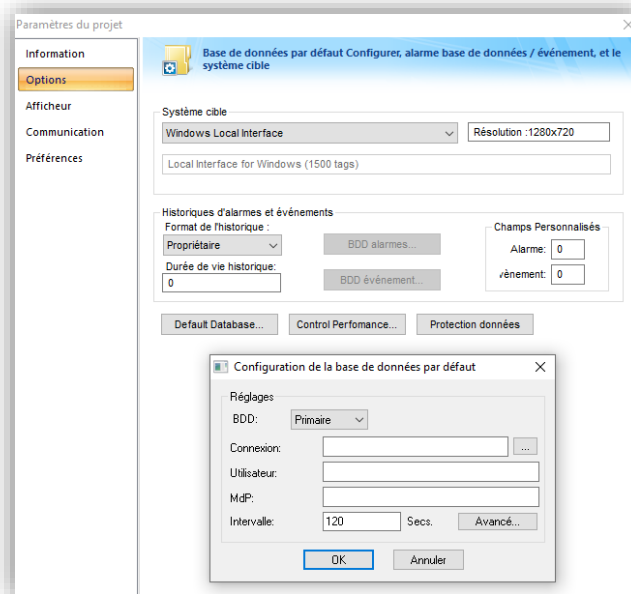


Figure 2 – Schéma redondance : état 2

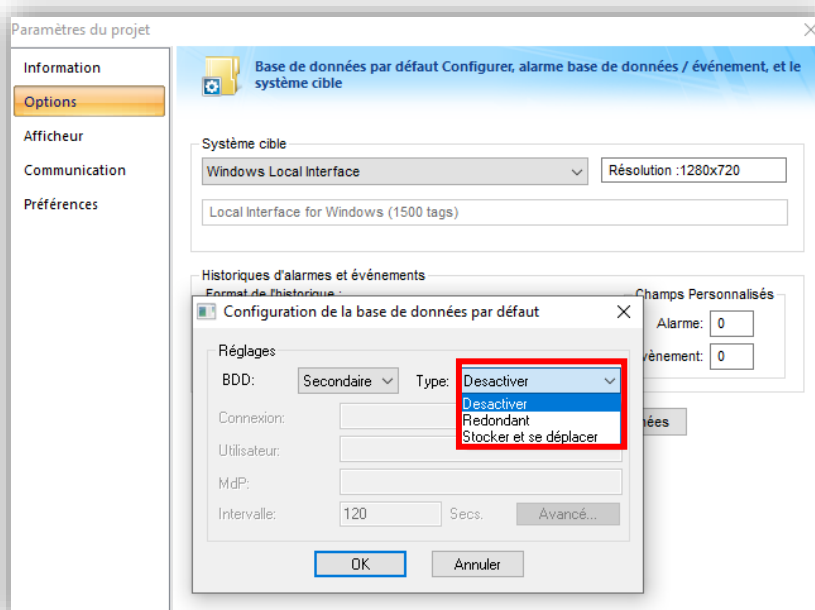
1- Redondance base de données

La redondance entre 2 bases de données se gère directement dans les paramètres du projet en définissant tout d'abord les paramètres de la base de données primaire :



Puis il faudra définir les paramètres de la base de données secondaire et choisir le type d'enregistrement parmi les 3 options :

- **Redondant** : Dans ce mode, IWS enregistre les données dans les bases de données principale et secondaire. Si l'une de ces bases de données n'est pas disponible, IWS continue à enregistrer les données uniquement dans la base de données disponible. Lorsque la base de données non disponible redevient disponible, IWS synchronise automatiquement les deux bases de données.
- **Stocker et se déplacer** : Dans ce mode, IWS enregistre les données dans la base de données primaire uniquement. Si la base de données principale devient indisponible, IWS enregistre les données dans la base de données secondaire. Lorsque la base de données primaire redevient disponible, IWS déplace les données de la base de données secondaire vers la base de données primaire.

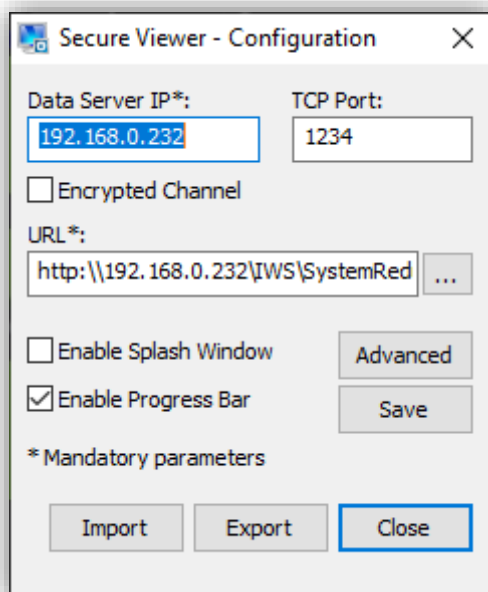


2- Redondance clients légers

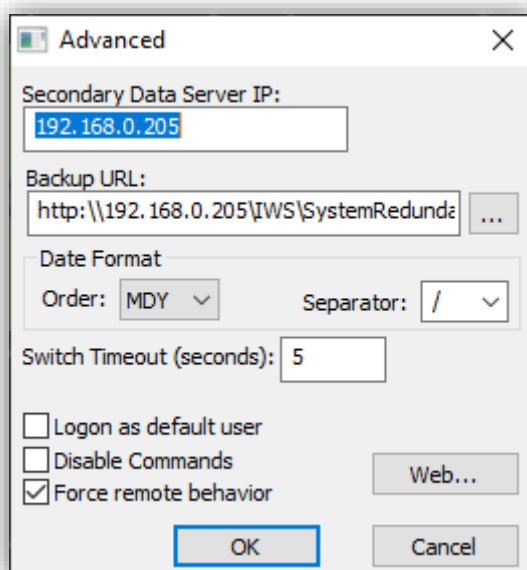
La redondance des clients légers permet d'automatiser la connexion d'un client léger vers le runtime qui sera accessible.

a) Secure viewer

Avec le client léger de type « secure viewer », la configuration se fait via l'utilitaire « secure viewer configuration ». Tout d'abord il faudra configurer la connexion au premier runtime en renseignant son adresse IP (Data Server IP) et le chemin vers le dossier web de l'application (URL) :



Puis la connexion au 2^e runtime avec les mêmes infos qui sont demandés. La case « force remote behavior » permet de forcer le comportement du client léger comme si il était sur une machine distante (utile uniquement si le client léger est installé sur la même machine que le runtime IWS) :



b) Client Web

Pour le client web la configuration de la redondance se fait de la même manière que pour le « Secure Viewer » mais cette fois-ci directement dans les paramètres du projet dans l'onglet Projet → Web :

The 'Paramètres du projet' dialog box is shown with the 'Web' tab selected. The title bar reads 'Paramètres du projet'. The main area has a blue header with a folder icon and the text 'Définir les paramètres de serveur et le client, tels que pare-feu et le clavier virtuel'. Below this, there are several settings:

- Adresse IP du serveur de données**: A text box containing an empty field, highlighted with a red rectangle.
- Période d'envoi (ms)**: A text box containing '100'.
- Canal Crypté**: ☐
- Interdire les Commandes de client dist**: ☐
- Mise à l'échelle auto**: ☒
- Autoriser Infobulles**: ☐
- Autoriser la compression de Fichier**: ☐
- Journal**:
 - Activer**: ☐
 - Nom de fichier**: A text box containing an empty field.
- Clavier Virtuel**: ☐
- Défaut**: A dropdown menu showing 'Keypad'.
- Infos**: A text box containing an empty field.
- Champs actif MIN/MAX**: ☐
- Activer les entrées texte m**: ☒
- échelle de dialogues Built-in**: A dropdown menu showing '100%'.

Buttons at the bottom: 'OK' (highlighted with a blue border) and 'Annuler'.

The 'Avancé' dialog box is shown. The title bar reads 'Avancé'. The main area has a light blue header with a folder icon and the text 'Définir les paramètres de serveur et le client, tels que pare-feu et le clavier virtuel'. Below this, there are several settings:

- Adresse IP du Serveur de données secondaire**: A text box containing an empty field, highlighted with a red rectangle.
- URL de sauvegarde**: A text box containing an empty field.
- ISSymbol URL**: A text box containing 'http://www.indusoft.com/ISSymbol/v81.4.0/'.
- Passerelle privée Web**:
 - Autorisé**: ☐
 - Port TCP**: ☒ 80
 - Port SSL**: ☐ 443
 - Adresse IP**: A text box containing an empty field.
 - Adresse IP secondaire**: A text box containing an empty field.
- Temps de réponse pour changement stations**: A text box containing '0'.

Buttons at the bottom: 'OK' (highlighted with a blue border) and 'Annuler'.

3- Redondance runtime et automate

Cette partie du document décrit les éléments à intégrer au développement de l'application pour la gestion de la redondance afin :

- De gérer automatiquement le basculement entre les 2 runtimes lorsque l'un des deux est arrêté ou inaccessible sur le réseau,
- De gérer automatiquement le basculement entre les 2 automates lorsque l'un des deux est arrêté ou inaccessible sur le réseau,
- D'avoir une application unique et similaire sur les 2 runtimes redondants,
- D'avoir un visuel sur l'état des runtime (savoir lequel est en cours d'utilisation, s'ils sont bien lancés, visualiser leurs adresses IP...).

L'application sera donc identique sur les 2 runtimes mais une seule communiquera avec les automates et la base de données. Si celui-ci s'arrête le 2^e runtime prendra la main automatiquement.

a) Paramètres

Tout d'abord un certain nombre de paramètres sont sauvegardés de manière permanente via la fonctionnalité « recette » :

- Adresse IP du runtime primaire,
- Adresse IP du runtime secondaire,
- Timeout avant le basculement vers le runtime secondaire,
- Adresse IP de l'automate primaire,
- Adresse IP de l'automate secondaire,
- Timeout avant le basculement vers l'automate secondaire.

	Nom du Tag	Nombre d'éléments
1	RedundancyPrimaryServerIP	
2	RedundancySecondaryServerIP	
3	RedundancyServerTimeout	
4	RedundancyPrimaryPLCStation	
5	RedundancySecondaryPLCStation	
6	RedundancyPLCTimeOut	

Ces paramètres seront chargés à chaque démarrage du runtime et modifiables via la vue « redundancy settings » dans l'application.

HMI/SCADA Stations

Primary Server IP: 192.168.0.143

Secondary Server IP: 192.168.0.205

Timeout: 3 seconds

PLCs

Primary PLC Station:

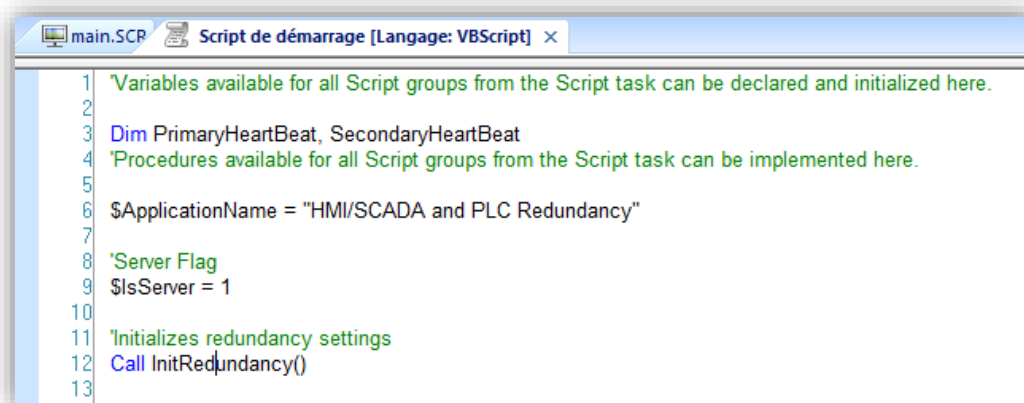
Secondary PLC Station:

Timeout: 3 seconds

Cancel Apply

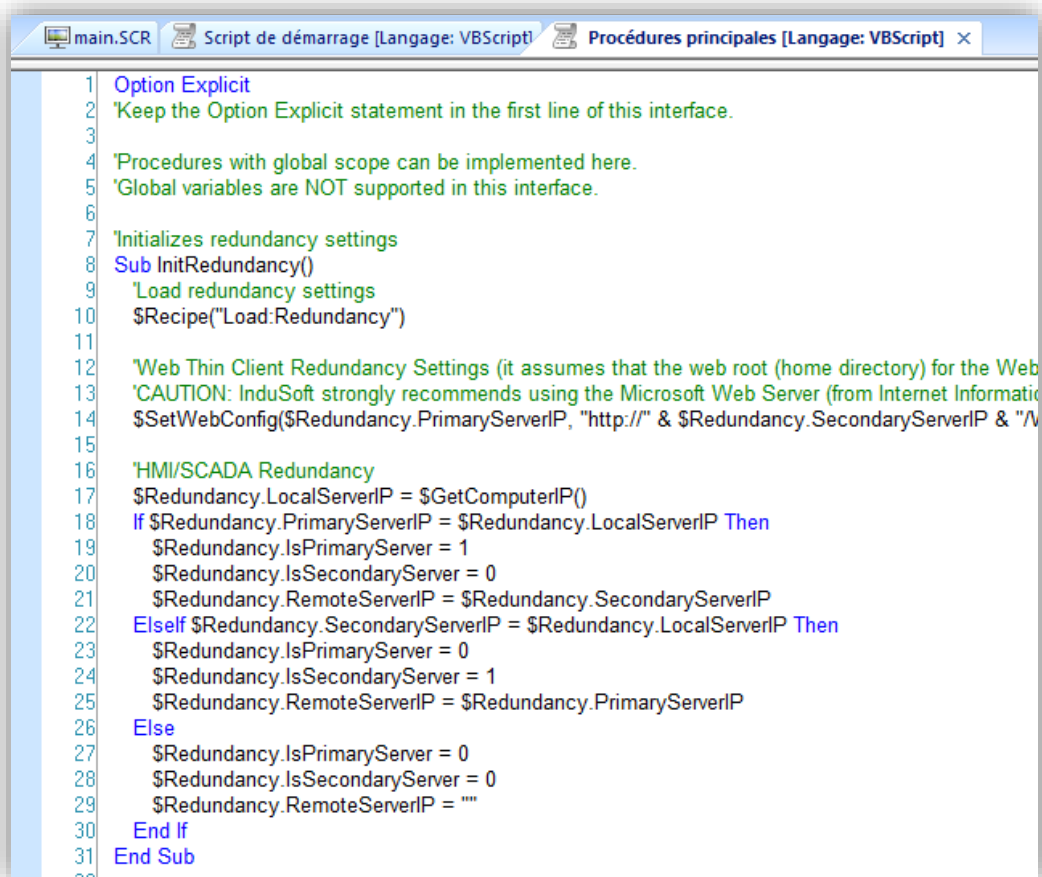
b) Initialisation

L'initialisation de la redondance commence par le script de démarrage :



```
1 'Variables available for all Script groups from the Script task can be declared and initialized here.
2
3 Dim PrimaryHeartBeat, SecondaryHeartBeat
4 'Procedures available for all Script groups from the Script task can be implemented here.
5
6 $ApplicationName = "HMI/SCADA and PLC Redundancy"
7
8 'Server Flag
9 $IsServer = 1
10
11 'Initializes redundancy settings
12 Call InitRedundancy()
13
```

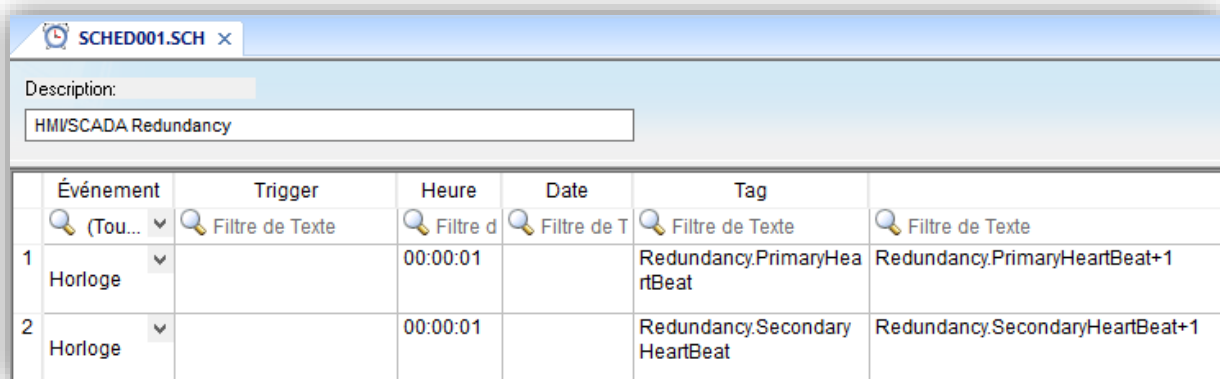
Ce script fait appel à une procédure qui initialise les paramètres de l'application en chargeant la recette :



```
1 Option Explicit
2 'Keep the Option Explicit statement in the first line of this interface.
3
4 'Procedures with global scope can be implemented here.
5 'Global variables are NOT supported in this interface.
6
7 'Initializes redundancy settings
8 Sub InitRedundancy()
9     'Load redundancy settings
10     $Recipe("Load:Redundancy")
11
12     'Web Thin Client Redundancy Settings (it assumes that the web root (home directory) for the Web
13     'CAUTION: InduSoft strongly recommends using the Microsoft Web Server (from Internet Information
14     $SetWebConfig($Redundancy.PrimaryServerIP, "http://" & $Redundancy.SecondaryServerIP & "/V
15
16     'HMI/SCADA Redundancy
17     $Redundancy.LocalServerIP = $GetComputerIP()
18     If $Redundancy.PrimaryServerIP = $Redundancy.LocalServerIP Then
19         $Redundancy.IsPrimaryServer = 1
20         $Redundancy.IsSecondaryServer = 0
21         $Redundancy.RemoteServerIP = $Redundancy.SecondaryServerIP
22     ElseIf $Redundancy.SecondaryServerIP = $Redundancy.LocalServerIP Then
23         $Redundancy.IsPrimaryServer = 0
24         $Redundancy.IsSecondaryServer = 1
25         $Redundancy.RemoteServerIP = $Redundancy.PrimaryServerIP
26     Else
27         $Redundancy.IsPrimaryServer = 0
28         $Redundancy.IsSecondaryServer = 0
29         $Redundancy.RemoteServerIP = ""
30     End If
31 End Sub
32
```

c) Tâches de fond

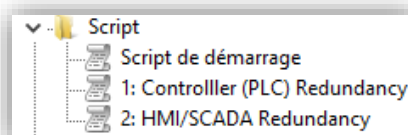
Parmi les éléments à intégrer en tâches de fond il y a une feuille de calendrier :



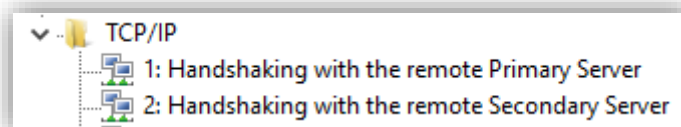
The screenshot shows a task scheduler window titled "SCHED001.SCH". Below the title bar is a "Description:" field containing "HMI/SCADA Redundancy". Below this is a table with the following columns: Événement, Trigger, Heure, Date, Tag, and Filtre de Texte. The table contains two rows of tasks.

	Événement	Trigger	Heure	Date	Tag	Filtre de Texte
1	Horloge		00:00:01		Redundancy.PrimaryHeartBeat	Redundancy.PrimaryHeartBeat+1
2	Horloge		00:00:01		Redundancy.SecondaryHeartBeat	Redundancy.SecondaryHeartBeat+1

Ainsi qu'un script qui tourne en continu et vient vérifier l'état de l'autre runtime (« HMI/SCADA Redundancy » dans le projet) :

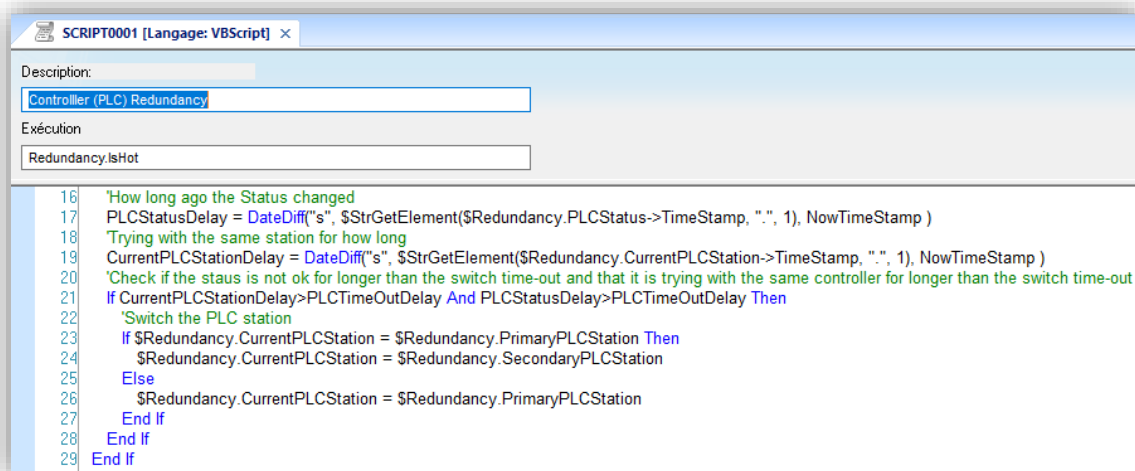


Deux feuilles de communication sont également créées dans le projet pour permettre d'échanger en TCP/IP sur l'état des runtimes. Les 2 feuilles ne sont pas activées simultanément, l'une est active uniquement si le runtime est actif, l'autre est active uniquement lorsque le runtime est en standby :



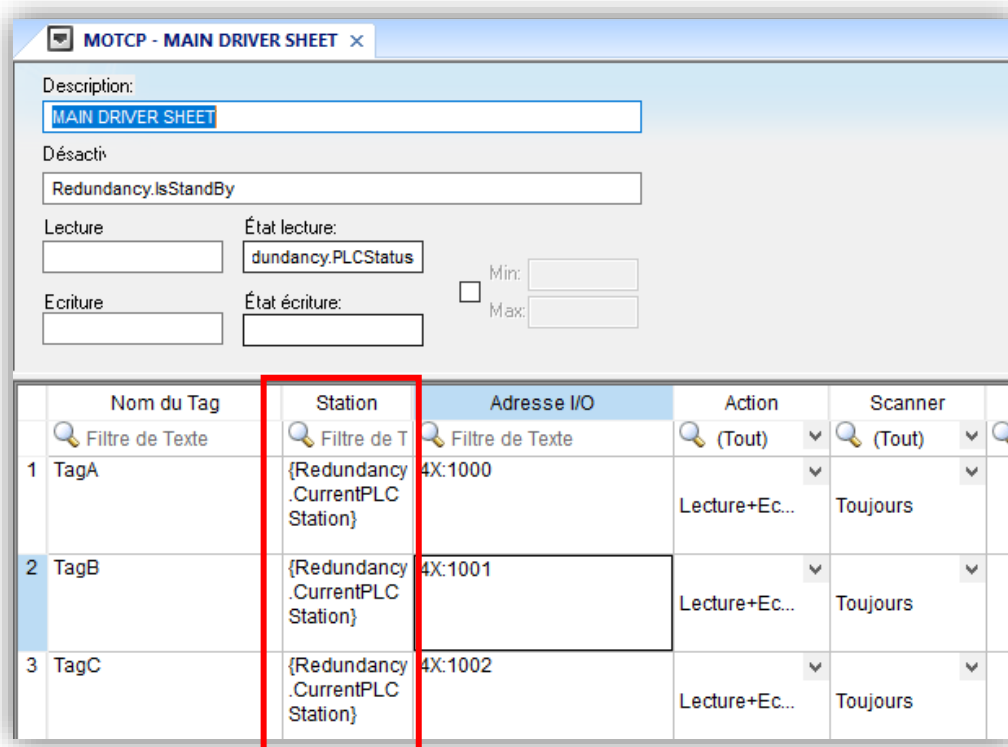
d) Redondance automates

Le script « Contrôleur (PLS) Redundancy » est actif uniquement pour le runtime IWS en cours d'utilisation et sert à contrôler que l'automate principal est bien actif. Dans le cas contraire les tags correspondants prendront automatiquement l'adresse de l'automate secondaire :



```
16 'How long ago the Status changed
17 PLCStatusDelay = DateDiff("s", $StrGetElement($Redundancy.PLCStatus->TimeStamp, ".", 1), NowTimeStamp)
18 'Trying with the same station for how long
19 CurrentPLCStationDelay = DateDiff("s", $StrGetElement($Redundancy.CurrentPLCStation->TimeStamp, ".", 1), NowTimeStamp)
20 'Check if the status is not ok for longer than the switch time-out and that it is trying with the same controller for longer than the switch time-out
21 If CurrentPLCStationDelay>PLCTimeOutDelay And PLCStatusDelay>PLCTimeOutDelay Then
22 'Switch the PLC station
23 If $Redundancy.CurrentPLCStation = $Redundancy.PrimaryPLCStation Then
24 $Redundancy.CurrentPLCStation = $Redundancy.SecondaryPLCStation
25 Else
26 $Redundancy.CurrentPLCStation = $Redundancy.PrimaryPLCStation
27 End If
28 End If
29 End If
```

Le tag « Redundancy.CurrentPLCStation » correspond donc à l'adresse de l'automate et sera à renseigner dans les feuilles de communication pour que le runtime pointe vers le bon automate :



	Nom du Tag	Station	Adresse I/O	Action	Scanner
1	TagA	{Redundancy.CurrentPLCStation}	4X:1000	Lecture+Ec...	Toujours
2	TagB	{Redundancy.CurrentPLCStation}	4X:1001	Lecture+Ec...	Toujours
3	TagC	{Redundancy.CurrentPLCStation}	4X:1002	Lecture+Ec...	Toujours

a) Développement

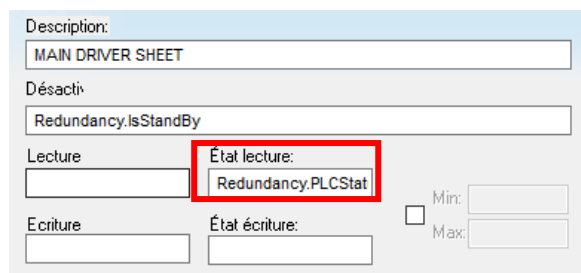
La gestion de la redondance telle qu'elle est gérée dans le projet permet grâce à certain tags de désactiver toutes les fonctionnalités sur le runtime secondaire qui est en standby.

Les tags en question sont les suivants :

- Redundancy.IsHot : ce tag est à 1 quand le runtime est celui qui communique avec les automates, base de données, ...
- Redundancy.IsStandBy : à l'inverse du tag précédent, ce tag est à 0 quand il communique avec les automates, bases de données, ...

Ils seront à renseigner dans les fonctionnalités suivantes afin de les désactiver dynamiquement si le runtime est en standby :

- **Feuilles de communication :**



Description: MAIN DRIVER SHEET

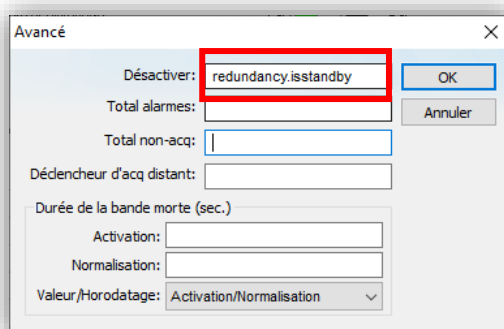
Désactiver: Redundancy.IsStandBy

Lecture: État lecture: Redundancy.PLCStat

Ecriture: État écriture:

Min: Max:

- **Feuilles d'alarmes :**



Avancé

Désactiver: redundancy.isstandby

Total alarmes:

Total non-acq:

Déclencheur d'acq distant:

Durée de la bande morte (sec.)

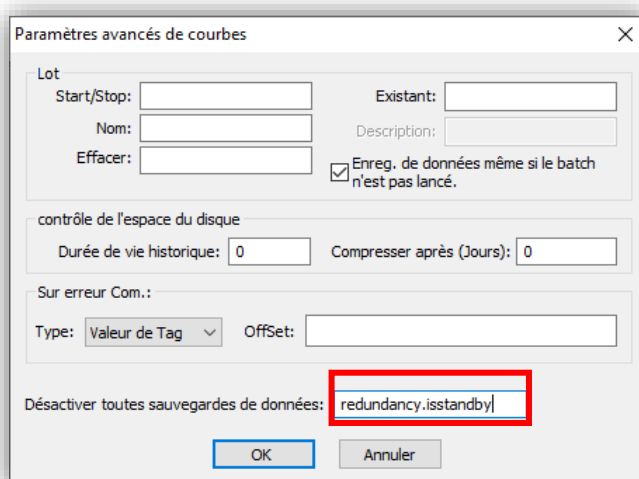
Activation:

Normalisation:

Valeur/Horodatage: Activation/Normalisation

OK Annuler

- **Feuilles de courbes**



Paramètres avancés de courbes

Lot

Start/Stop: Existant:

Nom: Description:

Effacer: Enreg. de données même si le batch n'est pas lancé.

contrôle de l'espace du disque

Durée de vie historique: 0 Compresser après (Jours): 0

Sur erreur Com.:

Type: Valeur de Tag Offset:

Désactiver toutes sauvegardes de données: redundancy.isstandby

OK Annuler

- **Scripts**

Description:
test
Exécution
redundancy.ishot

- **Logger d'évènement**

Régler les événements		X
☒ Activer journal D'evt	Désactiver:	redundancy.isstan BDD événement...
Réglages		
☐ Sécurité du système	☐ Recette	☐ Personnalisé Messages
☐ Navigation Ecrans	☐ Report	☐ Alerte de Système

Si les runtimes sont utilisés uniquement avec des clients légers il n'y aura rien d'autre à configurer, par contre si les viewers des runtimes sont utilisés il sera nécessaire de créer des feuilles TCP/IP supplémentaires pour échanger les différentes variables entre les 2 runtimes pour que le runtime en « standby » soit également utilisable.

b) Visuel

Dans l'application une page a été développée spécialement pour visualiser l'état des automates ainsi que des runtimes de manière claire :

