

Présentation de S7ProSim

S7ProSim fournit un objet COM qui donne accès par programme à l'interface de simulation de processus de S7-PLCSIM. Vous pouvez utiliser S7ProSim dans toute application qui peut accepter que des objets COM soient reliés à une simulation de processus S7-PLCSIM.

Ce document en ligne décrit comment ajouter S7ProSim à une application, ainsi que les caractéristiques, interface et fonctionnements de S7ProSim, notamment les définitions objets logicielles des méthodes et des événements.

Ajout d'un objet COM S7ProSim à votre projet

Pour ajouter un objet COM S7ProSim dans votre projet, vous insérez une référence à cet objet. Les étapes nécessaires pour ajouter une référence de projet dépendent de votre environnement de programmation. Dans Microsoft Visual Basic (6.0 ou .NET) par exemple, vous devez exécuter les étapes suivantes pour ajouter une référence d'objet COM S7ProSim :

1. Sélectionnez la commande **Projet > Références** ou **Projet > Ajouter référence**.
2. Dans la boîte de dialogue Références, cochez la case pour l'objet COM S7ProSim Siemens. Dans Visual Basic .NET, cette option se situe dans l'onglet COM de la boîte de dialogue Références.
3. Cliquez sur OK.

Une fois que vous avez ajouté la référence de projet, vous pouvez utiliser le navigateur d'objets pour examiner les méthodes et les événements de l'objet COM S7ProSim. Dans le navigateur d'objets, sélectionnez S7PROSIMLib dans la liste déroulante des bibliothèques. La classe S7ProSim contient les méthodes et les événements que vous pouvez utiliser pour programmer une interface vers S7-PLCSIM.

Dans Microsoft Visual Studio C++ V6.0 ou dans Microsoft Visual C++ .NET, suivez les procédures d'ajout d'objet COM qui sont pertinentes pour l'environnement de programmation concerné.

Programmation d'une interface vers S7-PLCSIM à l'aide de S7ProSim

Vous devez exécuter les tâches suivantes pour faire fonctionner l'automate de simulation S7-PLCSIM de manière programmée à l'aide de S7ProSim :

- Incluez l'objet COM Siemens S7ProSim dans le projet.
- Ajoutez une déclaration pour S7ProSim à votre projet.

Exemple : Visual Basic 6.0

```
Option Explicit
Private WithEvents S7ProSim As S7PROSIMLib.S7ProSim
...
Private Sub Form_Load()
Set S7ProSim = New S7PROSIMLIB.S7ProSim
...
End Sub
```

Exemple : Visual Basic .NET

```
Private WithEvents S7ProSim As New S7PROSIMLib.S7ProSim
```

Exemple : Visual C++ 6.0

```
// la bibliothèque/tlb ProSim est dans la dll
#import <S7wspsmx.dll> named_guids, no_namespace//interfaces brutes uniquement

class ProSimWrapper
{
public:
    ProSimWrapper() : m_pProSim(OLESTR("S7wspsmx.S7ProSim"), NULL,
CLSCTX_INPROC_SERVER)
    {}; // le pointeur intelligent est automatiquement créé
        // dans la pile au démarrage de l'application

    virtual ~ProSimWrapper()
    {}; // pas d'implémentation, le pointeur intelligent est
        // automatiquement libéré à l'arrêt de l'application

    IS7ProSim * GetPtr()
    {
        return m_pProSim;
    };

// Attributs
protected:
    // IProSimPtr est un pointeur intelligent CComPtr vers l'interface IProSim
    // Il vient du fichier dll de #import
    // CoCreateInstance sera appelé automatiquement dans l'objet
    // pointeur dans le constructeur de cette classe
    // la libération de pointeur (release ptr) est automatiquement
    // appelée par le destructeur de cette classe
    IS7ProSimPtr m_pProSim;
};
```

Exemple : C#

```
using S7PROSIMLib;
...
private S7ProSim ps;
```

- Pour Visual Basic, programmez des gestionnaires d'événements pour les événements S7ProSim. Les gestionnaires d'événements ne sont pas nécessaires dans Visual C++. Vous pouvez insérer, à l'intérieur de chaque gestionnaire d'événement, du code personnalisé pour votre application.

Exemple : Visual Basic 6.0

```
Private Sub S7ProSim_PauseStateChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_ScanFinished(ByVal ScanInfo As Variant)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_PLCSimStateChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_ConnectionError(ByVal ControlEngine As String, ByVal
error As Long)
    DoEvents
    MsgBox "Erreur de connexion"
End Sub

Private Sub S7ProSim_ScanModeChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub
```

Remarque

Dans Visual Basic .NET, l'appel "DoEvents" n'est pas nécessaire.

- Ajoutez des boutons de commande, des boîtes de texte et d'autres objets à votre application selon vos besoins pour accéder aux diverses méthodes S7ProSim. Programmez le code pour chaque gestionnaire de bouton de commande pour l'appel de méthodes S7ProSim et définissez les valeurs correspondantes pour les boîtes de texte en fonction de votre application.

Méthodes

◆ BeginScanNotify	Enregistre S7ProSim pour des rappels de l'automate Control Engine. L'événement ScanFinished et l'événement PLCSimStateChanged seront envoyés lorsqu'ils se produisent.
◆ Connect	Connecte S7ProSim à S7-PLCSIM.
◆ Continue	Poursuit une simulation qui a été suspendue.
◆ Disconnect	Déconnecte S7ProSim de S7-PLCSIM.
◆ EndScanNotify	Annule l'enregistrement de S7ProSim pour des rappels de l'automate Control Engine. Les événements ScanFinished et PLCSimStateChanged ne seront pas envoyés.
◆ ExecuteNmsScan	Force S7-PLCSIM à exécuter des cycles pendant une durée indiquée (Nms) et n'attend pas que l'exécution du cycle en cours soit achevée. Si la notification de cycle est activée, le programme sera averti lorsque S7-PLCSIM aura terminé les cycles.
◆ ExecuteNScans	Force S7-PLCSIM à exécuter un nombre défini de cycles sans attendre que l'exécution du cycle en cours s'achève. Si la notification de cycle est activée, le programme sera averti lorsque S7-PLCSIM aura terminé les cycles.
◆ ExecuteSingleScan	Force S7-PLCSIM à exécuter un cycle sans attendre que l'exécution du cycle en cours s'achève. Si la notification de cycle est activée, le programme sera averti lorsque S7-PLCSIM aura terminé le cycle.
◆ GetPauseState	Renvoie l'état Pause en cours de S7-PLCSIM.
◆ GetScanMode	Renvoie le mode d'exécution de S7-PLCSIM.
◆ GetStartUpSwitch	Fournit le paramétrage de mise en route (redémarrage, démarrage à chaud ou démarrage à froid) pour S7-PLCSIM.
◆ GetState	Renvoie une chaîne contenant la position en cours du commutateur à clé de S7-PLCSIM (position RUN, RUN-P ou STOP).
◆ HotStartWithSavedValues	Définit une valeur booléenne déterminant si S7-PLCSIM doit charger la périphérie d'E/S sauvegardée lors d'une mise en route à l'état de redémarrage HotStart. Pour que S7-PLCSIM démarre et charge la périphérie d'E/S, l'utilisateur doit appeler HotStartWithSavedValues avec la valeur TRUE, enregistrer le programme AP (SavePLC) et définir l'état de mise en route pour S7-PLCSIM à redémarrage HotStart (SetStartUpSwitch). Lors de son redémarrage, S7-PLCSIM chargera la périphérie d'E/S.
◆ Pause	Suspend une simulation.
◆ ReadDataBlockValue	Lit un bit, un octet, un mot ou un double mot particulier dans la zone de mémoire DB de S7-PLCSIM.
◆ ReadFlagValue	Lit un bit, un octet, un mot ou un double mot particulier dans la zone des mementos M de S7-PLCSIM.
◆ ReadOutputImage	Lit des éléments dans la périphérie de sortie (zone de mémoire PA) de S7-PLCSIM.
◆ ReadOutputPoint	Lit un bit (Boolean), un octet (Byte), un mot (Integer) ou un double mot (Long) spécifique de la périphérie de sortie (zone PA).

- ◆ **SavePLC** Enregistre les données en cours de l'AP de simulation dans un fichier.
Les données sauvegardées comprennent le programme, la configuration matérielle, la position du commutateur à clé indiquée par la fenêtre CPU, le mode d'exécution (cycle continu ou cycle unique), l'état des E/S, les valeurs des temporisations, les adresses symboliques, ainsi que le paramétrage de l'alimentation (sous tension ou hors tension).
- ◆ **SetScanMode** Définit le mode d'exécution pour S7-PLCSIM.
- ◆ **SetStartupSwitch** Définit le type de démarrage (redémarrage, démarrage à chaud ou démarrage à froid) à utiliser lors de la mise en route de S7-PLCSIM .
- ◆ **SetState** Définit la position en cours du commutateur à clé de S7-PLCSIM (RUN, RUN-P ou STOP).
- ◆ **StartPLCSim** Lance S7-PLCSIM avec le fichier de simulation AP indiqué (enregistré lors d'un appel précédent à SavePLC).
- ◆ **WriteDataBlockValue** Écrit un bit, un octet, un mot ou un double mot particulier dans la zone de mémoire DB de S7-PLCSIM.
- ◆ **WriteFlagValue** Écrit un bit, un octet, un mot ou un double mot particulier dans la zone des mementos M de S7-PLCSIM.
- ◆ **WriteInputImage** Écrit des éléments dans la périphérie d'entrée (zone PE) de S7-PLCSIM, en commençant à la position StartIndex des données désignées par pData.
- ◆ **WriteInputPoint** Écrit un bit (Boolean), un octet (Byte), un mot (Integer) ou un double mot (Long) spécifique du paramètre Data (de type Variant) dans la périphérie d'entrée (zone PE).

BeginScanNotify

STDMETHOD(**CS7ProSim::BeginScanNotify**)()

Description

Enregistre S7ProSim pour des rappels de l'automate Control Engine. L'événement ScanFinished et l'événement PLCSimStateChanged seront envoyés lorsqu'ils se produisent.

Paramètres

Néant

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.
PS_E_POWEROFF	0x80040212 : S7-PLCSIM est hors tension.

Utilisation Visual Basic

Function **BeginScanNotify**() As Long

Connect

STDMETHOD(**CS7ProSim::Connect**)()

Description

Connecte S7ProSim à S7-PLCSIM.

Paramètres

Néant

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.
PS_E_POWEROFF	0x80040212 : S7-PLCSIM est hors tension.

Utilisation Visual Basic

Function **Connect()** As Long

 **Continue**

```
STDMETHOD(CS7ProSim::Continue)()
```

 **Description**

Poursuit une simulation qui a été suspendue.

 **Paramètres**

Néant

 **Traitement des erreurs**

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

 **Codes retour**

Valeur	Signification
S_OK	0x00000000 : Réussite
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.

 **Utilisation Visual Basic**

```
Sub Continue()
```

Disconnect

STDMETHOD(**CS7ProSim::Disconnect**)()

Description

Déconnecte S7ProSim de S7-PLCSIM.

Paramètres

Néant

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_POWEROFF	0x80040212 : S7-PLCSIM est hors tension.

Utilisation Visual Basic

Function **Disconnect()** As Long

EndScanNotify

STDMETHOD (CS7ProSim::EndScanNotify) ()

Description

Annule l'enregistrement de S7ProSim pour des rappels de l'automate Control Engine. Les événements ScanFinished et PLCSimStateChanged ne seront pas envoyés.

Paramètres

Néant

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.
PS_E_POWEROFF	0x80040212 : S7-PLCSIM est hors tension.
PS_E_NOTREGISTERED	0x80040209 : S7ProSim n'est pas enregistré pour des rappels de S7-PLCSIM.

Utilisation Visual Basic

Function EndScanNotify() As Long

ExecuteNmsScan

STDMETHOD(**CS7ProSim::ExecuteNmsScan**)(long MsNumber)

Description

Force S7-PLCSIM à exécuter des cycles pendant une durée indiquée (Nms) et n'attend pas que l'exécution du cycle en cours soit achevée. Si la notification de cycle est activée, le programme sera averti lorsque S7-PLCSIM aura terminé les cycles. S7-PLCSIM doit être en mode Cycle unique pour que vous puissiez utiliser cette méthode.

Paramètres

 MsNumber Durée (en millisecondes) pendant laquelle les cycles doivent être exécutés.

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM n'est pas en mode d'exécution "cycle unique".
PS_E_PLCNORUNNING	0x8004020E : S7-PLCSIM n'est pas en cours d'exécution.
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.

Utilisation Visual Basic

Function **ExecuteNmsScan**(*MsNumber As Long*) As Long

ExecuteNScans

STDMETHOD(**CS7ProSim::ExecuteNScans**)(long NScanNumber)

Description

Force S7-PLCSIM à exécuter un nombre défini de cycles sans attendre que l'exécution du cycle en cours s'achève. Si la notification de cycle est activée, le programme sera averti lorsque S7-PLCSIM aura terminé les cycles. S7-PLCSIM doit être en mode Cycle unique pour que vous puissiez utiliser cette méthode.

Paramètres

 NScanNumber Nombre de cycles à exécuter

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM n'est pas en mode d'exécution "cycle unique".
PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM n'est pas en cours d'exécution.
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.

Utilisation Visual Basic

Function **ExecuteNScans**(*NScanNumber As Long*) As Long

ExecuteSingleScan

STDMETHOD (CS7ProSim::ExecuteSingleScan) ()

Description

Force S7-PLCSIM à exécuter un cycle sans attendre que l'exécution du cycle en cours s'achève. Si la notification de cycle est activée, le programme sera averti lorsque S7-PLCSIM aura terminé le cycle. S7-PLCSIM doit être en mode Cycle unique pour que vous puissiez utiliser cette méthode.

Paramètres

Néant

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
E_FAIL	0x80004005 : Erreur non précisée
PS_E_PLCNORUNNING	0x8004020E : S7-PLCSIM n'est pas en cours d'exécution.
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM n'est pas en mode d'exécution "cycle unique".
PS_E_MODENOTPOSSIBLE	0x8004020C : S7-PLCSIM n'a pas pu établir le mode d'exécution de cycle indiqué.

Utilisation Visual Basic

Function **ExecuteSingleScan()** As Long

GetPauseState

STDMETHOD(**CS7ProSim::GetPauseState**)(PauseStateConstants *pVal)

Description

Renvoie l'état Pause en cours de S7-PLCSIM.

Paramètres

 pVal Pointeur désignant l'état S7-PLCSIM renvoyé, qui est l'une des constantes PauseStateConstants

Remarques

En cas d'appel depuis Visual Basic, l'état Pause est renvoyé dans le code retour de la fonction et il n'y a pas de paramètre pVal.

En cas d'appel depuis C++, l'état est renvoyé dans la valeur désignée par pVal.

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.

Utilisation Visual Basic

Function **GetPauseState()** As **PauseStateConstants**

GetScanMode

STDMETHOD(**CS7ProSim::GetScanMode**)(ScanModeConstants *pVal)

Description

Renvoie le mode d'exécution de S7-PLCSIM.

Paramètres

 pVal Pointeur désignant le mode d'exécution renvoyé. Le mode d'exécution renvoyé est l'une des constantes ScanModeConstants

Remarques

En cas d'appel depuis Visual Basic, le mode d'exécution est renvoyé dans le code retour de la fonction et il n'y a pas de paramètre pVal.

En cas d'appel depuis C++, l'état est renvoyé dans la valeur désignée par pVal.

Traitement des erreurs

Les erreurs sont renvoyées dans l'événement ConnectionError, et non dans l'appel de la fonction.

Codes retour

Valeur	Signification
S_OK	0x00000000 : Réussite
PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.

Utilisation Visual Basic

Function **GetScanMode()** As **ScanModeConstants**

Événements

- ◆ **ConnectionError** Généré lorsque la connexion à l'automate Control Engine ("S7-PLCSIM") est impossible ou en cas d'erreur concernant l'appel de toute méthode S7ProSim.
- ◆ **PauseStateChanged** Généré lorsqu'un changement d'état Pause/Poursuivre est détecté. NewState est une chaîne représentant l'une des constantes PauseStateConstants.
- ◆ **PLCSimStateChanged** Généré lorsqu'un nouvel état du commutateur AP est détecté. NewState est le nouvel état de fonctionnement : "RUN", "RUN_P" ou "STOP".
- ◆ **ScanFinished** Généré lors de l'achèvement d'un cycle unique. ScanInfo fournit des informations indexées sur le cycle.
- ◆ **ScanModeChanged** Généré lorsqu'un changement de mode d'exécution ScanMode est détecté. NewState est une chaîne représentant l'une des constantes ScanModeConstants.

ConnectionError

HRESULT **ConnectionError**(BSTR *ControlEngine*, long *Error*)

Description

Généré lorsque la connexion à l'automate Control Engine ("S7-PLCSIM") est impossible ou en cas d'erreur concernant l'appel de toute méthode S7ProSim.

Utilisation Visual Basic

Event **ConnectionError**(*ControlEngine As String, Error As Long*)

 **PauseStateChanged**

HRESULT **PauseStateChanged**(BSTR NewState)

 **Description**

Généré lorsqu'un changement d'état Pause/Poursuivre est détecté. NewState est une chaîne représentant l'une des constantes PauseStateConstants.

 **Utilisation Visual Basic**

Event **PauseStateChanged**(*NewState As String*)

PLCSimStateChanged

HRESULT **PLCSimStateChanged**(BSTR NewState)

Description

Généré lorsqu'un nouvel état du commutateur AP est détecté. NewState est le nouvel état de fonctionnement : "RUN", "RUN_P" ou "STOP".

Utilisation Visual Basic

Event **PLCSimStateChanged**(*NewState As String*)

ScanFinished

HRESULT **ScanFinished**(VARIANT ScanInfo)

Description

Généré lors de l'achèvement d'un cycle unique. ScanInfo fournit des informations indexées sur le cycle.

Utilisation Visual Basic

Event **ScanFinished**(*ScanInfo*)

ScanModeChanged

HRESULT **ScanModeChanged**(BSTR NewState)

Description

Généré lorsqu'un changement de mode d'exécution ScanMode est détecté. NewState est une chaîne représentant l'une des constantes ScanModeConstants.

Utilisation Visual Basic

Event **ScanModeChanged**(*NewState As String*)

Définitions de types

 CPURunMode	Constantes pour l'état d'exécution de cycle de la CPU
 ImageDataTypeConstants	Constantes pour la méthode ReadOutputImage
 PauseStateConstants	Constantes pour l'état Pause
 PointDataTypeConstants	Constantes pour la méthode ReadOutputPoint
 RestartSwitchPosition	Constantes pour la position du commutateur de mise en route sur la face avant
 ScanModeConstants	Constantes pour le mode d'exécution
 tagPauseState	Constantes pour l'état Pause
 Constantes ScanInfo	Constantes pour des informations sur le cycle

CPURunMode

```
enum CPURunMode { CONTINUOUS_SCAN, SINGLE_SCAN, SINGLE_STEP }
```

Description

Constantes pour l'état d'exécution de cycle de la CPU

Éléments

CONTINUOUS_SCAN

SINGLE_SCAN

SINGLE_STEP

 **ImageDataTypeConstants**

```
enum {  
    S7Byte = 2,  
    S7Word = 3,  
    S7DoubleWord = 4  
}
```

 **Description**

Constantes pour la méthode ReadOutputImage

 **Éléments**

S7Byte
S7DoubleWord
S7Word

PauseStateConstants

```
enum {  
    Running = 0,  
    Paused = 1,  
    Disabled = 2  
}
```

Description

Constantes pour l'état Pause

Eléments

Disabled

Paused

Running

 **PointDataTypeConstants**

```
enum {  
    S7_Bit = 1,  
    S7_Byte = 2,  
    S7_Word = 3,  
    S7_DoubleWord = 4  
}
```

 **Description**

Constantes pour la méthode ReadOutputPoint

 **Éléments**

S7_Bit

S7_Byte

S7_DoubleWord

S7_Word

RestartSwitchPosition

```
enum {  
    WarmStart = 0,  
    HotStart = 1,  
    ColdStart = 2  
}
```

Description

Constantes pour la position du commutateur de mise en route sur la face avant

Eléments

ColdStart Remise en route avec l'OB102

HotStart Remise en route avec l'OB101

WarmStart Remise en route avec l'OB100

 **ScanModeConstants**

```
enum {  
    SingleScan = 0,  
    ContinuousScan = 1  
}
```

 **Description**

Constantes pour le mode d'exécution

 **Éléments**

ContinuousScan

SingleScan

tagPauseState

```
enum tagPauseState { ENABLED_RUNNING, ENABLED_PAUSED, DISABLED }
```

Description

Constantes pour l'état Pause

Eléments

DISABLED

ENABLED_PAUSED

ENABLED_RUNNING

Constantes ScanInfo

Constantes ScanInfo

 ScanInfo

 NUM_OF_SCANINFO_ELEMENTS	nombre d'éléments dans le tableau de retour ScanInfo
 EXECUTION_TIME_NDX	indice 0 : temps d'exécution en ms
 MIN_CYCLE_TIME_NDX	indice 1 : plus petite valeur de temps d'exécution en ms
 LARGEST_CYCLE_TIME_NDX	indice 2 : plus grande valeur de temps d'exécution en ms
 AVERAGE_CYCLE_TIME_NDX	indice 3 : temps de cycle moyen en ms
 IS_PLC_RUNNING_NDX	indice 4 : memento : 1='AP s'exécute ; 0='AP ne s'exécute pas.

 **ScanInfo**

ScanInfo

Le type de données Variant **ScanInfo** représente un tableau de doubles mots (Long). Chaque double mot dans le tableau spécifie une information sur le cycle, comme défini par les constantes ScanInfo.

NUM_OF_SCANINFO_ELEMENTS

```
#define NUM_OF_SCANINFO_ELEMENTS 5
```

Description

nombre d'éléments dans le tableau de retour ScanInfo

EXECUTION_TIME_NDX

```
#define EXECUTION_TIME_NDX 0
```

Description

indice 0 : temps d'exécution en ms

MIN_CYCLE_TIME_NDX

```
#define MIN_CYCLE_TIME_NDX 1
```

Description

indice 1 : plus petite valeur de temps d'exécution en ms

LARGEST_CYCLE_TIME_NDX

```
#define LARGEST_CYCLE_TIME_NDX 2
```

Description

indice 2 : plus grande valeur de temps d'exécution en ms

AVERAGE_CYCLE_TIME_NDX

```
#define AVERAGE_CYCLE_TIME_NDX 3
```

Description

indice 3 : temps de cycle moyen en ms

IS_PLC_RUNNING_NDX

```
#define IS_PLC_RUNNING_NDX 4
```

Description

indice 4 : memento : 1=!'AP s'exécute ; 0=!'AP ne s'exécute pas.

Codes d'erreur

❖ PS_E_BADBITNDX	0x80040205 : Indice de bit incorrect
❖ PS_E_BADBYTECOUNT	0x80040202 : La taille du tableau de données est incorrecte pour l'indice d'octet de début indiqué.
❖ PS_E_BADBYTENDX	0x80040201 : Indice d'octet incorrect
❖ PS_E_BADTYPE	0x80040206 : Type de données incorrect
❖ PS_E_INVALIDCALLBACK	0x80040207 : Rappel incorrect
❖ PS_E_INVALIDDISPATCH	0x80040208 : Distribution incorrecte
❖ PS_E_INVALIDINPUT	0x80040213 : Entrée incorrecte
❖ PS_E_INVALIDSCANTYPE	0x8004020B : Mode d'exécution incorrect ; doit être l'une des constantes ScanModeConstants
❖ PS_E_MODENOTPOSSIBLE	0x8004020C : S7-PLCSIM n'a pas pu établir le mode d'exécution de cycle indiqué.
❖ PS_E_NOTALLREADSWORKED	0x8004020F : Toutes les opérations de lecture n'ont pas abouti.
❖ PS_E_NOTALLWRITESWORKED	0x80040210 : Toutes les opérations d'écriture n'ont pas abouti.
❖ PS_E_NOTCONNECTED	0x80040211 : S7ProSim n'est pas connecté à S7-PLCSIM.
❖ PS_E_NOTIFICATION_EXIST	0x8004020D : S7ProSim est déjà enregistré pour la notification
❖ PS_E_NOTREGISTERED	0x80040209 : S7ProSim n'est pas enregistré pour des rappels de S7-PLCSIM.
❖ PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM n'est pas en mode d'exécution "cycle unique".
❖ PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM n'est pas en cours d'exécution.
❖ PS_E_POWEROFF	0x80040212 : S7-PLCSIM est hors tension.
❖ PS_E_READFAILED	0x80040203 : Echec de l'opération de lecture
❖ PS_E_WRITEFAILED	0x80040204 : Echec de l'opération d'écriture
☰ E_FAIL	0x80004005 : Erreur non précisée
☰ E_INVALID_STATE	0x00008002 : Etat incorrect
☰ S_OK	0x00000000 : Réussite
☰ STG_E_CANTSAVE	0x80030103 : Enregistrement impossible