PSDIRECT
CitectSCADA Driver for Siemens S5 and S7 and TI PLCs

User information and design

CitectSCADA

Driver version history

Version	Modified By	Details
2.00	JFP	2.0 is the Stable version after Initial development
2.01	JFP	Fixed problem with S5 LONGS Fixed S7 I PI & PE addresses Improved some sections & troubleshooting
2.2	JFP	• Added extra DCB validity checks everywhere & made sure of potential string overflows • Compiled against the CTDDK V6 (dont seem to be any major changes) • Added support for Siemens S7 "STRING" Datatype (new feature, not supported by S7NT)
2.2.1	JFP	Added S7-200 support to backend
2.2.2	JFP	Added some more FAQ's & updated status addresses
2.3	JFP	• Increased number of IO devices to 128 • Increased blocksize to CiTect max of 2048 bits, only useful for those using big arrays • Now allow multiple channels, but still recommend only one • New Status register addresses to monitor backend redundant address status • Now allow multiple Citect IO devices connecting to one backend IO Device • Substantial work on the IEC 870 protocol
2.4	JFP	• Fixed problem with S7 strings @ max block length • Fixed problem with S5 strings • Completely re-tested & re-vamped the TI profile • Added support for Square D driver
2.4.2	JFP	
2.4.3	JFP	Added some more troubleshooting items
2.5.14.4	JFP	Released with backend 7.10.54, added supportAdded notes for S7-1200
2.5.3	JFP	With frontend 2.5.3
2.6	JFP	Tidy up & merge added IEC870 chapter

Contents

1 User information

1.1 Application notes for PSDIRECT CitectSCADA Front-end Interface

Type	Detail
Manufacturer	www.ProSCADA.com
Device name	Siemens S5 and S7 and TI PLCs
Communications method	Direct DLL interface into driver poll cache

The following CitectSCADA Drivers may be replaced by this Driver

S7NT   - Siemens Softnet systems

S7NTSP   - Siemens misaligned real addressing

SINEC   - Siemens TF Systems for S5

Applicom   - Layer 4 for S5

TINECWIN,TINECNT, TIDIRECT – Simatic TI via CP1434 H1 card

SQUARED   - Square D Ethernet

1.2 Introduction

In large applications where CitectSCADA has to talk to many Siemens PLC's, a need has arisen to have a higher performance Siemens Driver interface. Simatic Net has a lot of limitations on the number of PLCs it can connect to and also the performance of the Siemens API has been an issue.   CitectSCADA also needed a way to optimise the way it communicates to the PLCs, particularly if the user attempted to poll large configurations of Function block instance DB's directly (User Defined types).

The PSDIRECT CitectSCADA direct interface driver is a front-end / back-end driver, meaning that the driver consists of two parts:

- The back-end server polls the data from the PLC's at configurable intervals. The back-end puts the data in a memory cache.

- The front-end is a native CitectSCADA driver that reads the data from the cache instantly with 0 delay (does not poll the PLC's).

1.3 Advantages over the traditional CitectSCADA Drivers

This has the following advantages:

1. The Polling of the data is completely **de-coupled** from the rate at which the clients are requesting it, which means the clients cannot choke up the IO server with requests - no matter how badly the system is configured.

2. With a traditional CitectSCADA Driver, all polling is Dynamic & client driven. With this driver **each block of data has an individual poll time**. This allows each block of data to be optimised to poll as SLOW as is acceptable for that sub-application. This then leaves more resources available for other data that needs to be faster. For example: poll analogs at 5 second interval, setpoints & run-hours at 30 second interval. **Operator perception of speed** is associated to digital status and alarm inputs. These can be polled at 0.5 seconds intervals.

3. The data is already available when a client polls it and data is displayed immediately without delay. For example popup screens open instantly with data from the cache.

4. The back-end polls raw blocks of data from the PLC. The front-end can then request any data type from this one polled block. This results in substantial efficiencies when you have many mixed data types in close address proximity. For example: Siemens Instance DB's (User Defined types) like the PID loop DB's are very complicated structures containing floats, words & bits. The CitectSCADA S7NT driver would generate a poll for each data type whereas this driver would read the information in **one network poll** block of bytes to be stored in the cache, the Front end driver then processes a **CitectSCADA Driver request for each data type** from the same cached data.

5. Communications can be configured and debugged outside CitectSCADA.

1.4 Advantages over the Siemens Softnet Solutions

This section refers to the Siemens Ethernet back-end.

To setup Siemens Simatic Net on a PC and to test the communication can be very tedious. Multiple IO servers have to be used to communicate to more than 20 PLC effectively.

The PSDIRECT driver will take 5 minutes to install. The driver works with any standard 10MB, 100MB or 1GB Ethernet card. To date more than 50 PLCs on a single system have been tested with one standard 100Mb card. Multiple Ethernet cards can be used to communicate to PLCs, however there are no real performance or reliability gains. The driver works equally well sharing CitectSCADA, Windows workgroups, & Siemens ISO thru 1 Ethernet card. A good Ethernet switch handles different link speeds & network node isolation & redundancy.

The PLC's are the bottlenecks & the total Ethernet bandwidth used by the Ethernet card is < 5% typically**.** This driver maximises the communication to all the PLC's on you network by implementing a multithreaded parallel polling strategy.

This driver supports up to 128 PLC's on one Network. It has been proven on a system with 50 PLC's & 80 000 IO.

2 Structure of the driver

The different protocol levels are displayed in the table below:

Citect32.EXE
PSDirect.DLL (standard CiTect driver)
<xxx>STSR.DLL (depends on protocols)
Driver Poll Cache Memory mapped file
SIXSPOLL.EXE (back end polling)
Siemens H1 ISO Transport OR TCP/IP

The flow of information in the driver starts at the network protocol "Siemens H1 ISO Transport". This protocol is supplied with the driver. No Siemens hardware or software is required at the CitectSCADA side.

The Polling of Blocks of data from the PLC is performed by the hidden application <xxx>SPOLL.EXE. This application is started whenever client applications request data. On start-up, the last configuration file used is opened. The application copies the configuration file into a memory mapped file (MMF/CACHE) and then updates all polled data to the MMF.

The tag interface DLL SIXSTSR.DLL converts IO address strings into internal references to find the data in the MMF.

The PSDirect.DLL is a standard CitectSCADA Array block driver which reads the tags using Native PLC address references. This means you can configure an IO Server with a minimal "Comms" project and run different tag projects on the Display Clients as per normal CitectSCADA conventions.

3   Setup guide

3.1   Step 1 Install the PSDIRECT CitectSCADA front-end driver

This is provided as a standard CitectSCADA Driver install package, just run the Setup.exe provided and make sure you choose the correct install directory for the version of CitectSCADA.

**Note After installing, It is recommended that you copy CitectSCADA\Bin\PROTDIR.DBF to both the user\<project>\Include AND the main or Comms Projects. (ref KB Q2546)

3.2   Step 2 – Install the Back-end Driver

Install the Siemens Ethernet Driver by running the setup.exe provided.

The recommended install location is the **Citect\Bin** Directory.

Recommended Program Group on the Start menu is "CitectSCADA".

3.3   Step 3 – Install "Siemens H1 ISO transport" (***Not used for TCP/IP)

*This step is <u>**optional**</u>, depending on the protocols used to talk to the PLC's, if you are using TCP/IP you DO NOT Require the ISO transport – Please go to Section 4 Setting up the driver.

See Installing "Siemens H1 ISO transport" (***Not used for TCP/IP)

4   Setting Up the driver

Run the "Siemens Industrial Ethernet Driver Configurator"

The Default configuration file will open, this includes 3 PLC's an S5, S7 & TI.

Since you will probably be using an S7 select the S5 PLC & press delete, do the same for the TI PLC

The S7 example configuration is setup to poll M0-M99; this address exists in all S7 PLC's so it should poll correctly without setting up a Step 7 program on the PLC.

Before setting addresses etc. please set the protocol type in the channel settings

- Open the channel settings to set the protocol – press the button

You have a choice between TCP/IP and ISO. If ISO is chosen, then the ISO transport must be installed (supplied with the driver).

- Set the PLC's address, for ISO this is a 12 digit Hexadecimal MAC address, for TCP/IP this is of the form 0.0.0.0

- Now press ok to go back to the main device settings

The read TSAP setting defaults to "R=0;S=2" which means read the CPU in Rack 0 Slot 2. This is Correct for an S7-300. On a S7-400 with 2 slot power supply, meaning the CPU is in Slot 3, this should be R=0;S=3. **Note the Slot is the slot number of the CPU, NOT the CP 443 card!

***For the S7-200 PLC Set the read TSAP to "CP243" because the S7-200 does not comply with the rack & slot system.

Now "Reload" the configuration by pressing the button. This is IMPORTANT; always reload any configuration change changes. The configurator works on the configuration file on disk, the reload command loads that into memory, briefly interrupting the polling as it does so

Now check that you have communications. The transmit and receive counters at the bottom of the window should increment

The Status of the M0 block should change to a ✔

4.1   Setting Up using the serial MPI

The Serial S7 Driver uses a different Backend driver to the Ethernet however the configuration is almost identical. The main difference is the Channel settings where you setup the com port & baud rates.   The Local MPI address is normally 1, if you have 2 SCADA's on the network then you need to set SCADA a to 1 & SCADAB to 2 (No PLC may then use 2) & enable the Multi master setting.

5 Quick start Simple CitectSCADA configuration

Follow the following steps to configure a CitectSCADA application to the sample above.

** there is an example CitectSCADA project installed with the PSDIRECT Front end driver in "C:\Program Files\Citect\CitectSCADA\PSDIRECT Sample"

Read more about it in The Appendices at the end of this document. The following is a simple Fresh setup.

- Create a new CitectSCADA Project
- DO NOT create a **Boards** object
- Add a **Port** & Call it "PS_PORT" (set address to 0, all other fields blank incl. the board)
- Add an IO device call it "S7PLC" connect to the port "PS_PORT"
- Set the address to "S7;Ethernet" or "S7;MPI" for serial
- Set the Protocol to "PSDIRECT1"
- Add a Tag with IO device "S7PLC" & IO Address "M0" data type set to INT.
- Put the tag on a mimic; compile the application and start CitectSCADA.
- Verify the correct data is displayed on the mimic.

6 Frequently asked questions & troubleshooting

Q: What is this ?
A: the driver's "channel/network" settings, select protocol, timouts

Q: What is this enable ?
A: Backup channel is only applicable to ISO protocol see redundancy

Q: What is this enable ?
A: enables backup address

Q: What is the Backup device connection?
A: see the section on redundancy

Q: What are the meaning of these symbols?
A: see the FAQ below, Green is Good

Q: What is exception type?
A: not used with citect

Q: What's the meaning of this symbol ?
A: The block is offscan, data not in use

Q: What is Access time?
A: When Citect stops accessing the data the block goes offscan after this time

Q: When the driver is active, more and more messages appear. Are these messages errors or is it just a message about writing to the DBs. What does (0) en (1) mean?
A: These messages indicate write actions, Generally an important indication of what the SCADA is doing.   It is our experience that many SCADA developers are unaware how often their systems write, which affects performance.   We recommend you write your Cicode scripts to first read the variable, then only write new value if new value is different. The (1) means this data is written along with (0) in one packet

Q: The latest version 7.9.54 has PLC options for S7-300 & S7-400, what has changed since I note that my old configurations still work & have defaulted to S7-300

A: The protocol is identical, the S7-400 can handle larger block sizes than the S7-300 and is therefore faster – so it is a performance optimisation.

Q: I have REAL or LONG tags where some are displaying correctly, others not

A: This is a standard CiTect driver issue, CiTect always reads arrays of data so if your variables are not aligned to each other on 4 byte boundaries, the misaligned tags get invalid data (**Note this may be random, some work, some don't – usually tag debug works). For example DB12,10 - DB12,18 are 4 byte aligned, however DB12,18-DB12,32 are not. The driver has built in an option to read data in blocks of 32 bits which normally solves this issue. To apply this simply change the IO device protocol from PSDIRECT1 to PSDIRECT6 then recompile (remember to recompile the clients). The only disadvantage is that you loose STRING[ ]s & arrays which need to be on PSDIRECT1 on a separate IO device   :-(

Q: What do the Icons in the configurator mean ?

A:

✔	The block is healthy & contains current data
✖	Communications failure
🕑	The block is currently not in use, it is off poll, all the blocks on a standby server should be in this state (clock with green slash)
❓	This block is newly configured & is not currently being polled. You need to reload the configuration to activate. **note if Citect uses this address it will cause the NIO-*Address errors until configuration reload

Q: The driver configurator is using all my PC's CPU time and the whole machine is slow, what is wrong ?

A: You have many error messages flooding the bottom window of the configurator, this scrolling is consuming CPU time

- Pull the horizontal splitter down to close the bottom window, this reduces CPU time

- Use "Menu>Options>Learn addresses" to solve the NIO-*Address… errors, then driver reload

Q: Why do I see messages like "Nio-*Address Error DB15,W248" in the bottom window of the configurator ?

A: This means that CiTect is reading that address however this address is not configured for polling in the backend.

- **It is Crucial to Fix this issue**, before doing any other testing

- Note that this issue may occur with I & Q addresses where you have many small discontigous poll blocks in the backend. What is occurring is that Citect is aggregating address blocks. The only possible solution is to use the small blocksize version of the front end driver- this means changing the CiTect IO device protocol from PSDIRECT1 to PSDIRECT6. This change may still not eliminate the reporting of the errors however CiTect will get valid data.

- You only see these errors when CiTect tries to access the data, so seldom used addresses may not be noticed.

- There are 2 Semi automatic methods of solving this

    o By selecting Menu>Options>Learn addresses, the configurator uses those error messages to stretch existing blocks or add new ones.

    o By selecting Menu>Options>Import Addresses. The configurator can add / stretch blocks by using a VARIABLE.DBF or VARIABLE.RDB file. This works better than the learn address method because you get ALL the addresses in the project. If you open a UNITS.DBF file it will add IO Devices to the backend.

***Note this only changes the edited file in disk you need to "reload" to make it active in the Background polling program. Then you should see fewer address errors, you then need to Learn addresses again. Read more

Q: Why do I see "Nio- Address Error DB15,W248" in the bottom window when DB15,W248 is not used in any tag.

A: You have tags above & below DB15,W248 and the CiTect block optimiser is aggregating them. The PSDIRECT Driver is unable to figure out which addresses are used & which not. You have to build backend blocks spanning the entire DB not missing a single byte. This may be a problem with I & Q addresses, which cannot span unconfigured IO. A Partial solution is to use the 32 bit PSDIRECT6 protocol.

Q: I am unable to get communication, what are the most common problems?

A: If using ISO protocol Check that the Siemens ISO transport is installed, if you see the message in the bottom window "failed to open ISO channel 1" then the ISO transport is not correctly installed OR there may be a problem in the registry and the driver is not using the correct Ethernet card.

The read TSAP setting defaults to "R=0;S=2" which means read the CPU in Rack 0 Slot 2. This is Correct for an S7-300. On a S7-400 with 2 slot power supply, meaning the CPU is in Slot 3, this should be R=0;S=3. **Note the Slot is the slot number of the CPU, NOT the CP 443 card!

If you have 2 CitectSCADA's communicating to a S7-300 the third CitectSCADA connection will fail. By default the S7-300 has 8 communication channels – 2 for step 7, 2 for SCADA, 4 for PLC-PLC, This configuration can be changed in later model CPU's. The S7-400 seldom encounters this problem because it has 8 connections

Q: How do I update the configuration after making changes ? Or Why do newly added blocks show a status

A: Remember the configurator is not actually polling the PLC's The background polling program sixspoll.exe is doing that. Save the file then Select "File|Reload" off the menu, or press ▣. This saves the configuration and tells the background polling program to stop & reload it's configuration into memory. During this process you may see #COM in Citect.

Q: I have problems reading Direct S7 IO (I,Q,PI etc.) what do I do ?

A: You need to make sure that the poll blocks exactly match the configuration in the PLC. The PLC WILL NOT allow you to poll across an address for which no IO cards exist. You need to break this up into smaller blocks. **Note Citect Does not recommend writing directly to outputs for safety reasons.

Q: How do I setup the IO server to enable redundant CitectSCADA IO Servers ?

A: The driver behaves like all Citect drivers in this regard eg. The same as S7NT.

- Set the INI file parameter "AlwaysReturnNoError" = 0 (this is default, so this parameter need not exist)

- The poll block S0->S199 must be defined in the driver backend poll configuration. Being a frontend / backend driver reading from it needs status information about the communincation status of the backend, these variables are in the S addresses and are used internally for Unit Status by the front end

- Now if the driver loses communication to the PLC the Unit status will go bad & CitectSCADA's redundancy system will switch over to the backup IO Server

- **Note The "AlwaysReturnNoError"=1 is intended for debugging only. When set you will see correct values only if the driver is successfully polling, if not INVALID data may be displayed & no redundancy failovers will occur.

Q: My communications are Slow, WHY ?

A: Please Note the following, typical performance achievable

- This driver polls as fast as the PLC allows, the PLC is the bottleneck(slow side), the figures below are per PLC, the driver performance scales linearly per PLC.

- S7-315 CPU typically achieves 15-20 polls per second (We believe it is crippled)

- S7-416 CPU typically achieves upto 150 polls per second (10 x faster). This is with 2 IO Devices talking to the same PLC.

- A Vipa PLC typically achieves upto 40 polls per second with default settings, if communications resources are increased this can reach 100+

- The driver has a Built in Limit of 200 polls per second per IO Device, so in some cases you can get better performance by splitting your data into 2 IO Devices. However we do not recommend this as, the spare PLC resources should be reserved for Writing data & PLC-PLC communications & other Citect IO servers.

- From version 7.9.52 performance has been increased substantially over the above numbers the difference is more noticeable for configurations of lots of small blocks

Q: The communications are Slow, how do I make it faster ?

A: read the section about optimisation of the driver & try some of the following modifications at the PLC side.

Q: How do I configure communications to an S5 PLC ?

A: This is far more complicated than an S7 PLC. Please read the Driver help file (SIXHLP.CHM) section "Setting up the hardware" for Full instructions. In addition there are example S5 PLC programs & CP card setup files in the backend distribution in the "PLC" directory.

Q: I keep getting an error "Nio-niomap() different from map popup box ?

A: You are using more than one IO Channel device on the IO server, please put all IO Devices including different PLC types under one single Citect IO Channel object. **Note as of PSDIRECT 2.3 it is more tolerant of multiple channels.

Q: With the S7NT driver objects like, PID set point sliders operate smoothly, however with PSDIRECT it sometimes briefly jumps back to old position before assuming new position.

A: The PSDIRECT is actually faster here !!!, remember the Display is reading from the PSDRIECT cache & displaying at the Citect screen refresh rate, with S7NT it was actually stalling the screen refresh until the readback occurred.

Q   I have redundant PLC's can I configure redundancy at a lower level than the Citect level ? ie use a single Citect IO device but 2 IP addresses in the backend.

       OR Does the driver support the S7-400H hotstandby PLC ?

A: Yes simply configure the second IP addresses & rack & slot in the Backup fields, read more in Configuring for Redundancy

Q: How do I debug the operation of the low level backend redundancy

A: read the section on Using the Special Debug addresses

Q: Is the PSDIRECT Configurator designed to import addresses from arrays?. E.g. in Citect you can define an integer tag array with an address that looks something like DB10,0[20]. However the import functionality does not work with arrays in the back-end.

A: Arrays are supported, the data has to be polled in the backend, so if DB10,0[20] is a LONG or REAL (all examples assume 32bit) the poll block has to be at least DB10,0 to DB10,79. As of Backend 7.9.51.005 the database import now supports Arrays. You can however solve this problem online in seconds during runtime.

- if you open the Driver configurator while Citect is running & open a page with address references that are not currently being polled, you will see in the bottom window messages like

  NIO-*Address error DB10,D4

  NIO-*Address error DB10,D8

  ...

- Now off the options menu select, "Learn Addresses", you should see DB10,0 to DB10,3 change to DB10,0 to DB10,79 (you may have to go to another PLC & back)

- What it has done is use the NIO-* messages to stretch the block, this configuration then needs to be made active by "Reloading the configuration" . During reload the driver stops & has to re-start which causes momentary #COM but when it stabilises again you should be polling all data and the process only takes seconds of work. Read more

Q: Can the driver communicate with ProfInet ?

A: Technically NO, however all ProfInet capable CPU's communicate S7 Functions thru the same Ethernet port so it does not matter ! The driver will work, remember the driver communicates with the PLC program NOT IO.

The PSDIRECT driver uses a protocol called "S7 Function" or GET/SEND of Object oriented large chunks of data (Instance DB's) and is processed in the PLC asynchronously in 10+(variable) millsecond rates, this is more appropriate between a SCADA & PLC. ProfInet is more about remote IO and processing that in sub millisecond interrupt driven processing.

In this sense your ProfInet remote IO is more appropriately handled by the PLC program. It is dangerous to use the PLC as a gateway to directly access the IO.

7 Reference: Required components

7.1 CitectSCADA

- CitectSCADA V5.41 or higher.
- CitectSCADA V5.50 has the most installations at this time.
- CitectSCADA V6.0 has been successfully tested
- CitectSCADA V6.1 has been successfully tested
- CitectSCADA V7.0 has been successfully tested
- CitectSCADA V7.2 has been successfully tested
- CitectSCADA V7.4 has been successfully tested
- CitectSCADA V7.5 has been successfully tested
- CitectSCADA V8.0 has been successfully tested
- CitectSCADA V8.1 has been successfully tested
-

7.2 Operating Systems

- Windows 9x, Millennium NOT SUPPORTED for ISO protocol
- Windows NT4 SP4 or later
- Windows 2000 (Pro or Server)
- Windows XP SP1&2
- Windows 2003 Server / R2
- Windows Vista
- Windows 2008 Server / R2 (***NB at this time 64 bit systems cannot use ISO)
- Windows 7
- Windows 8
- Windows 2012 Server
- Windows 10
- Windows 2016 Server

7.3 Recommended System

CiTect recommendations

7.4 Network Card

Any NDIS compatible card should work. Named brands are recommended for best performance.

*** NOTE NO SIEMENS SOFTWARE OR HARDWARE REQUIRED

***NOTE when using the ISO protocol DO NOT install any Siemens software on the machine.

***NOTE Some problems have been encountered with Broadcom cards, recommend Intel

8 Reference: Communications Forms

8.1 Boards Form

This driver does not require a BOARDS object for Later versions of CiTect. Some people have reported problems omitting the board on this driver if they use other drivers that require boards.

**Note Adding The Board object WILL NOT cause problems, however you may see error messages at startup, these can be ignored.

8.2 Ports Form (channel)

Note: Unlike the S7NT driver, **only 1 port is compulsory on the IO server**; performance is just as good with only 1 because it is reading instantly from a cache. Please note this includes the case where you have both S7 & S5 PLC's Only use one Channel. If you use Both the Ethernet & Serial drivers put the serial devices on a separate port.

**Note from V2.3 the driver no longer reports errors when multiple channels are used, however we still recommend using only 1 channel

Field	Default	Allowable values
Port Name	This field is user defined and is not used by the driver.	
Port number	0	Any value, other than ""
Board name	Leave blank	Board is not required
Baud rate	Leave blank	
Data bits	Leave blank	
Stop bits	Leave blank	
Parity	Leave blank	
Special Opt	Leave blank	
Comment	This field is user defined and is not used by the driver.	

8.3 I/O Devices Form

The Address AND Protocol refers to the Address format of the tags & also specifies the type, the combination is used to decide which Driver DLL to load & what address format to Use. The following table lists the allowable combinations.

PROTOCOL	Address	Comment
PSDIRECT1	S7;ETHERNET[;N=xx]	Siemens Industrial Ethernet S7 PLC's
PSDIRECT1	S7;MPI[;N=xx]	Siemens Serial MPI
PSDIRECT2	S5;ETHERNET[;N=xx]	Siemens Industrial Ethernet S5 PLC's
PSDIRECT3	Conet[;N=xx]	OmniFlex Conet with timestamping
PSDIRECT4	TI[;N=xx]	Texas Instruments H1
PSDIRECT5	IEC;<common addr>[;N=name]	IEC 870-5-101, with common address
PSDIRECT6	S7;ETHERNET[;N=name]	Same as PSDIRECT1 with 32 bit blocksize. S7NTSP replacement
PSDIRECT6	S7;MPI[;N=name]	Same as PSDIRECT1 with 32 bit blocksize. S7NTSP replacement
PSDIRECT7	Blank	Square D emulates SQUARED
PSDIRECT8	Blank	GE Aritech Fire Panel
PSDIRECT1	S7;ETHERNET;STAT;N=name	Use to read Status variables that never show #COMM
PSDIRECT2	S5;SERIAL[;N=name]	S5 using 3964R serial protocol

The **Address** field is used to specify which backend driver to use & which protocol. The ";N=xx" is [optional] (do not type [ ]) and if this parameter is supplied the xx must be the IO device name in the backend.

The **Name** of the IO device is by default used to connect to the backend device of the same name. Unless the ";N=xx" parameter is supplied in which case THIS overrides the name. The ";N=xx" may therefore be used if IO device names Longer than 11 characters are required OR Multiple Citect IO devices connected to 1 backend IO device are required.

8.4 Driver Caching

This driver has its own cache, it is therefore not necessary to cache data on CitectSCADA side. The recommended cache enabled setting is **FALSE** (default is blank-which is TRUE). This setting is not critical, but performance will be at least 300 ms faster than the defaults.

8.5   I/O Devices form settings

Field	Default	Allowable values
Name	Set to the same as the Device name in the Driver Configurator Unless a ";N=xx" is supplied in which case the xx is used instead.	
Number	Must be blank or unique, (used by citect for redundancy).	
Address	**See above**	Specifies addressing & back-end driver
Protocol	**PSDIRECT<n>**	See table above
Port name	Refers to the port previously defined in 'ports' form.	
Comment	This field is user defined and is not used by the driver.	
Enable Caching	**FALSE**	
Cache time	Not used – no cacheing300 but set cache to FALSE	

8.6   Reference: Data types

Note: This driver is designed to mimic other CitectSCADA drivers & should supply all the same addresses & data types as the equivalent CitectSCADA driver does.

***Note the PSDIRECT6 32bit blocking protocol does not support STRINGS & Arrays

8.6.1 Reference – Tag Addressing

This driver supports many different Addressing formats depending on the device in use.

8.6.1.1 Siemens S7 Addressing

The S7 addressing is designed to be compatible with that used by the Standard Citect S7NT Driver.

Address Format	CitectHMI/SCADA Data Type
DB<no>,<index>[.<bitno>]	Digital, Byte, INT, BCD, LONG, LONG_BCD, REAL, STRING
DI<no>,<index>[.<bitno>]	Digital, Byte, INT, BCD, LONG, LONG_BCD, REAL, STRING
A<index>[.<bitno>]	INT, BYTE, Digital
Q<index>[.<bitno>]	INT, BYTE, Digital
E<index>[.<bitno>]	INT, BYTE, Digital
I<index>[.<bitno>]	INT, BYTE, Digital
M<index>[.<bitno>]	Digital, Byte, INT, BCD, LONG, LONG_BCD, REAL, STRING
PA<index>[.<bitno>]	INT, BYTE, Digital
PQ<index>[.<bitno>]	INT, BYTE, Digital
PE<index>.<bitno>	INT, BYTE, Digital
PI<index>.<bitno>	INT, BYTE, Digital
T<index>(10ms)	BCD (PLC s5time format)
T<index>(100ms)	BCD (PLC s5time format)
T<index>(1s)	BCD (PLC s5time format)
T<index>(10s)	BCD (PLC s5time format)
Z<index>	BCD
TDB<no>,<index >	REAL,INT (PLC s5time format - See note 1 below)
TDI<no>,<index >	REAL,INT (PLC s5time format - See note 1 below)
TM<index >	REAL,INT (PLC s5time format - See note 1 below)
SDB<no>,<index>[string len]	Siemens STRING datatype (See note 5 below), string len compulsory
V<index>[.<bitno>]	S7-200 V memmory area see S7200

Syntax:

The syntax is defined as follows (case insensitive)

DB<no>   <index>

DI<no>   <index>.<bitno>

Where:

DB or DI   data block or instance block

<no>   number of the data block or instance block

<area>A   output

Q   output

E   input

I   input

M   bit memory

PE   peripheral input

PI   peripheral input

PA   peripheral output

PQ   peripheral output

<index>   element number relative to start of block

<bitno>   bit within the element number

NOTES:

1. Addressing format T, TDB, TDI, TM: When one of these address formats is used, the driver will convert a time value (from CiTect HMI/SCADA Real to Siemens S5Time or vice versa) using the smallest time base it can fit in. Range: 0 - 9990.00 seconds. (0 <= Values < 0.01 will be truncated to 0 and Values > 9990.01 will generate an Alarm). Precision may suffer in the conversion, e.g. 10.01 will be converted to 10.00 (1100 S5Time format).   **Note having 1 timer value in a DB recommended datatype is REAL. However for >1 you MUST set datatype to INT this means you loose subseconds (unfortunately)

2. Addressing: DB, DI, A, E, M, PA , PE, TDB, TDI, and TM are using byte addressing. T, C, and Z are using word (two bytes) addressing. They are the same as in STEP7.

3. STRING data type: CiTect HMI/SCADA STRING data type is not the same as Siemens' STRING data type. It is equivalent to Siemens' CHAR Array data type. A NULL terminator (Siemens' BYTE data type, value = 0) is required by CiTect. If Siemens' STRING needs to be accessed, use the SDB Address.

4. the TDB & TM S5 timer datatypes are supported, however note the following caveat. If the tag datatype is INT the value returned is in seconds, the address in the PLC is a 16 bit int and arrays function normally. If the datatype is REAL the value is also in seconds & subseconds but addresses must be on 4 byte boundaries – skipping blank word addresses in the PLC CitectSCADA S7 Drivers.

5. S7 STRING datatype is supported using the SDB address. The values may be written from CiTect & the string length byte (in the PLC) will be set by the driver. The string will only read/write upto the max string len(in the PLC).**There are serious limitations however**

    a. ONLY 1 STRING is supported per DB, or at least block size apart (1st byte & last byte of adjacent strings must be >256 bytes apart).

    b. The CiTect array size MUST be < 127 bytes. Always set CiTect Array size==PLC string len

6. The following addresses are not supported:

- T   Siemens Timer
- C   Siemens Counter

*Note: Siemens Timer & counters in TM & TDB memory IS supported

**An Additional Address the SW addresses is available for status & debugging please read [Using the Special Debug addresses](#)

8.6.1.2 Siemens S5 Addressing

The Siemens S5 Addressing is designed to emulate the "SIEMENS.DBF" protocol. This may differ from the TF/Sinec or applicom addressing. The address format is almost identical to that used in the Step 5 programming software.

Data Types	Address Format	CitectHMI/SCADA Data Type
DATA WORDS	D\<data block>:\<word>	BCD / DIGITAL / INT / LONG / LONGBCD / REAL / STRING
DIGITALS	D\<data block>:\<word>.\<bit>	DIGITAL
DATA WORDS	DX\<data block>:\<word>	BCD / DIGITAL / INT / LONG / LONGBCD / REAL / STRING
DIGITALS	DX\<data block>:\<word>.\<bit>	DIGITAL
Timer	TDB\<data block>:\<word>	INT,REAL

EXAMPLES:

Data Type	DIGITAL
Address	D010:001.14
Comment	Digital – Data Block Number 10 : Word Number 1.Bit Number 14
Data Type	INT
Address	D0010:002
Comment	Data Word - Block Number 10 : Word Number 2

8.6.1.2.1   Note on converting SINEC addresses

The Sinec TF system has an extra layer of abstraction that names memory objects to TF names. Normally in CiTect projects these memory variables are arrays of 16 bit words. To convert these addresses:

- Look in the TF configuration and find the Data block address of each TF array

- Search & replace the VARIABLE.DBF file with \<TF array name> replace with D\<dbn>:\<add DB offset to TF array index>.\<bit>

8.7   Driver reference

	Detail
Driver name	PSDIRECT
Maximum array size PSDIRECT1	2048 Bits
Maximum array size PSDIRECT6	32 Bits

8.8   Driver generated error codes

- PROSCADA_ERR_NO_TAGS   0x100
- PROSCADA_ERR_BAD_OID   0x101
- PROSCADA_COMMS_ERROR   0x102
- PROSCADA_BAD_DATA_TYPE   0x103
- PROSCADA_UNSUPORTED_CITECT_TYPE   0x104

- PROSCADA_BAD_IOADDRESS   0x105
- PROSCADA_BAD_SIGNALCOND   0x106
- PROSCADA_BAD_HWOPT   0x107
- PROSCADA_DLL_LOAD_FAIL   0x108
- PROSCADA_WRITE_FAIL   0x109
- PROSCADA_ISO_NOT_CONNECTED   0x10A
  - Backend is not connected to the PLC – comms fail, unit offline
- PROSCADA_ADDRESS_ERROR   0x10B
  - Address does not exist on backend
- PROSCADA_UNDEFINED_PROTOCOL   0x10C
  - The Unit address settings were not recognised
- PROSCADA_HEARTBEAT_FAILED   0x10D
  - PLC is in Stop mode OR heartbeat monitoring is enabled but the variable is not changing.
- PROSCADA_BACKEND_NOT_RUNNING   0x10E
  - The backend driver SIXSPOLL.EXE has stopped running

8.9 Parameters, options, and settings

8.9.1 Standard Parameters

Parameter	Default	Allowable values
Block (bits)	2048	8-2048 (PSDIRECT6=32)
Delay (mS)	30000	1000-60000
MaxPending	8	Not critical with this driver, it replies instantly
Polltime (mS)	1000	Not critical but do not change, used for unit status
Timeout (mS)	Not used	
Retry	Not used	
WatchTime (Sec)	10	(rate at which offline IO devices are checked)

8.9.2 Driver Specific Parameters

Parameter	Default	Allowable values
AlwaysReturnNoError	0	0 Should be selected for Backup failover & displaying #COM on screen. It is useful to set this to 1 when testing with no PLC then you can use the learn address feature even if you have no PLC.
ValueIfCommsErr	0	Value to display on screen if Communications to PLC fail, set to something like '888" to make it obvious

HeartBeatAddress_<devname>	No Hb	The S7 driver continues polling normally when the PLC is in Stop mode resulting in frozen values on the screen. To force #COM when the PLC is in stop mode requires configuration of Heartbeat monitoring read more about it in Heart beat monitoring section of this document

To add these parameters to the CITECT.INI file use the following format

[PSDIRECT]

AlwaysReturnNoError=1

**Note setting the AlwaysReturnNoError to 1 like this is only useful for testing or engineering purposes.

9   Analysis & Trouble Shooting

Things to do on a call out to the IO server:

- Check to see if there are any popup dialogs from CitectSCADA or the Driver indicating a problem, if so write down the message displayed.

- Open the kernel & View the Driver's page

- Check the "Driver Errors" & "Timeouts" counters

- If these two counters are incrementing then CitectSCADA variables are addressing variables not configured in the back-end driver.

- Check which IO devices (PLC's) are offline in CitectSCADA

- Open the CitectSCADA kernel and go to the IO Servers page

- The statuses on the left indicate what is happening on this IO server, the statuses on the right indicate the status of the IO Server currently in use.

10   Using the Special Debug addresses

This driver is a cached Front End / Back End driver, so the standard CitectSCADA driver statistics are not very useful. This section describes the Special Tag addresses available for debugging purposes. All datatypes are INT

ST0   Connection   current connection number an internal value, this is normally static, is only meaning full when backup addresses are used

ST2   Conn State   0 means trying to connect to PLC, 2 means Connected

ST24   Reconnections   Increments every time the driver attempts to reconnect after the connection failure.   If this is going up, you have a bad network

ST26 Pollrecs per sec   Poll rate for this PLC - useful for checking optimisation. The rate shown at the bottom status block of the configurator should be the sum of these.

ST28 Disconnect Req's   Increments when PLC sends Disconnect Request. This Means that your Network is OK but the CP143 setup is not. The PLC is refusing connection requests.

ST32 PDU size   PLC request block size, 240 or 480 bytes

ST40 Primary connection   Connection number

ST42 Primary connection state, =2 means connected

ST80   Backup connection   Connection number of backup connection

ST82 backup connection state, =2 means connected

10.1.1 Redundancy control

When using the Backup IP address, the following logic may be used to indicate which connection is active.

ST0= =ST40 means that the Primary IP address is being polled

ST0= =ST80 means that the Backup IP address is being polled

Writing 1 to ST0 causes the driver to connect to the Backup connection

Writing 2 to ST0 causes the driver to connect to the Primary connection

10.2 For the S5 & TI PLC the Status addresses are as follows

ST1   Conn State   0 means trying to connect to PLC, 2 means Connected

ST12   Reconnections   Increments every time the driver attempts to reconnect after the connection a failed.   If this is going up, you have a bad network

ST13 Pollrecs per sec   Poll rate for this PLC - useful for checking optimisation. The rate in mission control should be the sum of these.

ST14 Disconnect Req's   Increments when PLC sends Disconnect Request. This Means that your Network is OK but the CP143 setup is not. The PLC is refusing connection requests.

ST21 Write Connection state. 2 means connected

11 Migration from S7NT driver systems

This is a brief summary of the steps required to migrate an existing S7NT system to the PSDIRECT Ethernet driver.

**Note Citect's behaviour is undefined if you forget to compile ALL clients to PSDIRECT, to avoid this issue change the IO Device numbers as part of the upgrade.

11.1 Communication forms

- No Boards form required
- **Delete all but 1 ports** set all fields to blank, address to "0"
- In IO devices (UNITS.DBF)
    - Make sure name is less than 11 characters (limitation of backend)
    - Set port name to the 1 port
    - Set Protocol to "PSDIRECT1" or "PSDIRECT6" for S7NTSP
    - Set Address to "S7;Ethernet"

11.2 Back-end Configuration

11.2.1 Backend IO devices

The Back-end configuration may be edited in bulk using excel & File Save As *.CSV. Save the default configuration to .CSV, and then edit it in Excel and copy data from the UNITS.DBF file above.

- From the Options Menu Select "Import Variables"
- In the file dialog select the user\<comms project> directory and open UNITS.DBF
- If you have a redundant CiTect system it will have imported 2 devices per PLC, delete all duplicates.
- Make sure each Device has a ST0 to ST200 IO block THIS IS NOT A PLC ADDRESS – it is internal driver status used by the front end to check unit status.
- You will need to manually set the network Addresses for each PLC
- Remember to Save & Reload to make the configuration active

11.2.2 Backend IO Blocks

Each CiTect tag must be contained inside a backend poll block. The data type of each block is "BYTE" any CiTect data type tag can be read from these blocks. If CiTect attempts to read from an address that is not currently polled you will get an NIO* IO address error in the bottom window of the configurator. There are 2 methods of semi automatically creating these blocks.

1. From the Options menu select "Import Variables"

    - In the file dialog select the user\<root project> directory and open VARIABLE.DBF or _VARIABL.RDB
    - Remember to Save & Reload to make the configuration active

2. After creating the Backend devices you can remove any address errors by selecting "Learn Addresses" from the options menu. This only functions while CiTect is running and the Address errors are flooding the bottom window.

    - Note that to make the new added, or stretched blocks active you need to save & reload the configuration.

- When this is done there may still be address errors, select "Learn Addresses" again & reload again.

- This process is not perfect you may still have to edit some manually

To achieve optimum performance you may still need to optimise the poll times of each block (see next section). You should at least sort the poll blocks in excel to tidy up.

12   Communication Optimisation

12.1 *note on performance in new version 7.9.52

This version can reads multiple block simultaneously, and when upgrading you should notice an improvement in the backend driver poll rate. This new version requires no additional changes, it will open existing configuration files. To take full advantage of this the only change to the configuration required is to set the PLC type to "S7-400" in the PLC type.   The greatest performance improvement will be seen with configurations with large numbers of small poll blocks.

12.2 How to Measure Driver Performance

- Create a minimal backend configuration with 1 poll block, 1 variable and a poll time of 0.0 seconds

- Note how many polls per sec you get in the configurator status at the bottom

- After measuring the speed, you can calculate the update time of your full configuration. e.g. If you get 10 pps (pollblocks per second), you can have

  - 1 pollblock updating 10 times a second or

  - 10 pollblocks updating at 1.0 seconds

  - 5 pollblocks updating at 1.0 seconds + 10 pollblocks updating at 2 seconds

If you configure more pollblocks, no errors will occur but you are not actually achieving your configuration update time. In this case the driver just processes each block in turn. This is not a problem most of the time but as you add more blocks your system will slow down & if you need faster update on a specific block, changing its polltime will achieve little.

12.3 Optimisation Techniques

To make a limited bandwidth system appear faster, you have to **slow down** the updating of less critical data to give more bandwidth to the critical data. (Increasing the pollrate of a block when the system is already at full stretch will have no effect) The basic principle is - poll only the data that the operator needs for a given picture & at the minimum rate he is prepared to accept. It is usually better to set slow polltimes at the start of a project which leaves plenty of room to expand, this sets a lower expectation and if you need more speed later you have spare bandwidth resources.

- Setpoints, Outputs, controls or any value that never changes in the PLC (only changes when modified from the SCADA) can be setup so that they hardly use any channel bandwidth at all. Set the polltime to a large value like 60 seconds. If a value is written the driver automatically forces the block to update so the user does not notice the slow poll times.

- Prioritize your data; find out how fast the data changes in the PLC. Do not poll faster than necessary especially while you are constructing your system. You can always speed up certain blocks later once your system is fully configured. Generally analog values can be polled slower e.g. 5 sec but Digital Inputs may need 1 sec.

12.4 PLC Optimisation

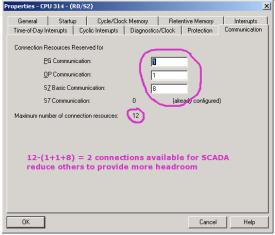### 12.4.1 *notes on PLC Performance

Siemens seem to cripple the performance of the S7-300. It is not unusual to get 10x performance improvement with the S7-400 versus the S7-300

Try to increase the PLC scan cycle by moving code to the Idle processing OB's OR process half the code each scan & thereby doubling the scan cycle.

12.4.2 Number of Network connection resources(CPU properties)

Resources available for the driver on the S7-300 is Two (2) by default, which can be changed this means 2 IO servers can connect but the third will fail, unless you change this configuration in Step 7. The S7-400 allows 4-8 connections by default.
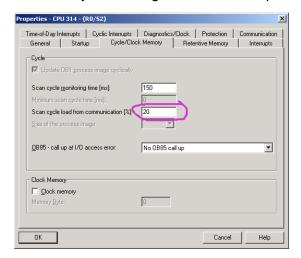
*note it is recommended that you increase this to leave plenty spare because sometimes the PLC holds onto old connections – blocking the driver's reconnection attempts.

**Note - by default the driver uses SCADA connections, which are the missing/unlisted connections in the picture above. The S7 Basic connections are PLC-PLC connections normally you can reduce this number. Ideally you want the SCADA connections free to be at least 2+the number of SCADA's. To force the driver to use an OP or PG connection set the local TSAP to "R=0;S=2;OP" or "R=0;S=2;PG"

12.4.3 CPU communication load priority (CPU properties)

You can try increasing communication priority

12.4.4 Block size (CP card properties)

Some CP cards (not all) have an option in the options tab for large PDU block size (>240) – set this option if available.

13   Using the Stress test example project

An example CitectSCADA project installed with the Front end driver in

C:\Program Files\Citect\CitectSCADA\PSDIRECT Sample\PROSCADA_test.ctz

To use this setup you need

- An S7 PLC 400 or 300 with Ethernet CP card

- Step 7 with Siemens NCM to configure the PLC CP card

- A CitectSCADA machine networked to the PLC, NO SIEMENS SOFTWARE Required

Probably the easiest protocol to use is TCP/IP

- Setup the PLC with an IP address and ping to test from the CitectSCADA

- NO Step 7 program is required or if one exists it MAY NOT modify m0-m511 OR simply put the PLC in STOP mode.

- Set the backend driver's channel protocol to TCP/IP

- Set the device "S7PLC" 's IP address

- File|reload the configuration

- Check that you have transmits & receives and the status of the blocks goes good

- Import the project into CitectSCADA, compile & run

- Open test page 1

The stress test writes huge amounts of data and reads it back comparing it to that written, any mismatches increment the error counters.

14   Setting Up Heartbeat monitoring

The S7 driver continues polling normally when the PLC is in Stop mode resulting in frozen values on the screen. To force #COM's when the PLC is in stop mode requires configuration of the Heartbeat monitoring feature..

For each IO device (Unit) configured, a parameter of the following form is entered into the Citect.INI file

HeartBeatAddress_<devname>=<address>

Where <devname> is the name of the IO device in CitectSCADA and in the backend driver

And <address> is a PLC address of a variable that MUST change very 5 seconds at least if the PLC is running. A simple way of achieving this is to increment an integer in OB1 of the PLC scan cycle. ** the address format here is the same as the OPC server so you specify W for Integer & D for longs etc.
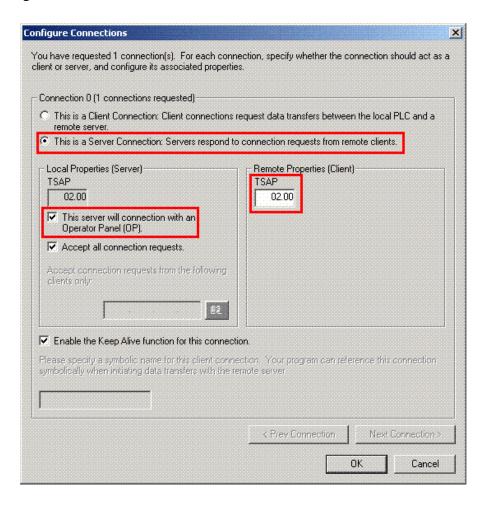
Examples

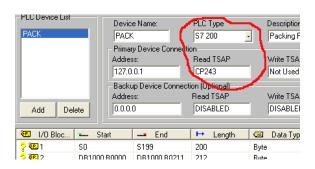HeartBeatAddress_S7PLC=MW0

HeartBeatAddress_PumpHouse=DB100,W44

HeartBeatAddress_Furnace2=DB130,D44 (long integer)

15   S7 200 Setup

1/ Run the Micro Win Ethernet Wizard
2/ we recommend that the CP243 be installed in position 0
3/ Set the CP243's Ethernet address
4/ Configure a New connection & set it as follows

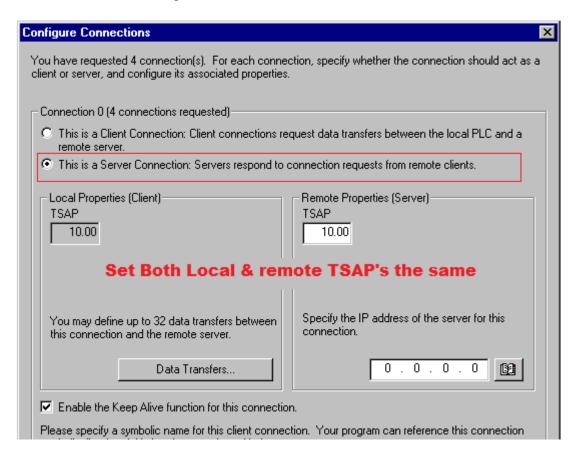In the Driver polling configuration set the Read TSAP field to "CP243"

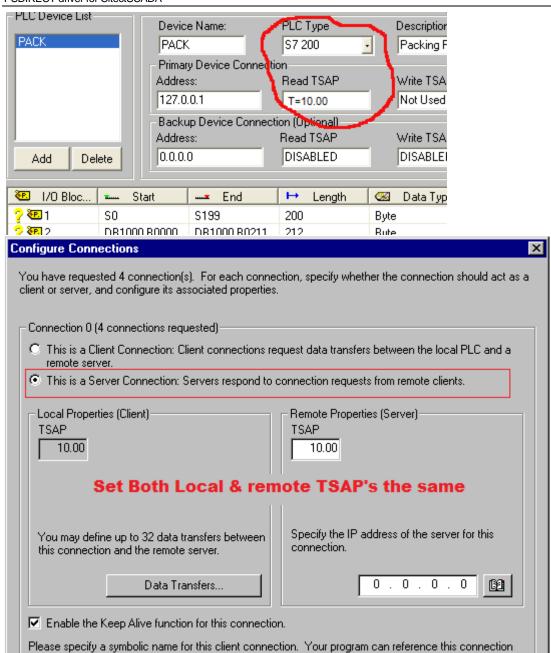You can poll the following data from an S7-200

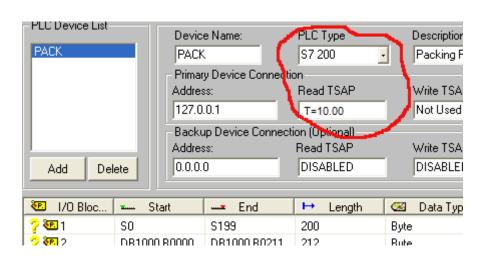M0-M32 (S7-200 has very small M area)
DB1,0->x is equivalent to the "V" mem area e.g. VW8 == CiTect block type INT address DB1,8
I0 len 2 (Built in I inputs)
Q1 len 1 (Built in Q outputs)

Copyright www.proscada.com

Alternative S7-200 configuration

Configure Connections

You have requested 4 connection(s). For each connection, specify whether the connection should act as a client or server, and configure its associated properties.

Connection 0 (4 connections requested)

○ This is a Client Connection: Client connections request data transfers between the local PLC and a remote server.

● This is a Server Connection: Servers respond to connection requests from remote clients.

Local Properties (Client)
TSAP
10.00

Remote Properties (Server)
TSAP
10.00

Set Both Local & remote TSAP's the same

You may define up to 32 data transfers between this connection and the remote server.

Data Transfers...

Specify the IP address of the server for this connection.

0 . 0 . 0 . 0

☑ Enable the Keep Alive function for this connection.

Please specify a symbolic name for this client connection. Your program can reference this connection
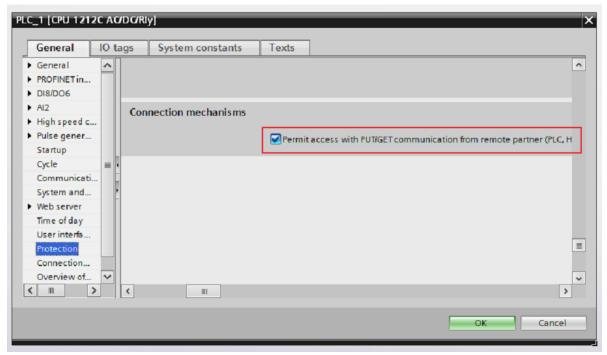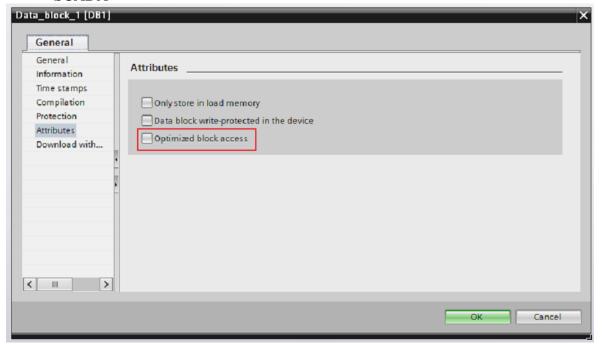
16   S7 1200 & 1500 Setup

In TIA-Portal PLC Setup check the following:
- CPU -> Properties -> Protection; make sure you TICK the "Permit access with PUT/GET communication from remote partner" and the protection itself must be set to "full access"

- UN-TICK the property "optimized block access" attribute on the DB's you want to read in SCADA

On the driver side:
- Set TSAP to R=0;S=1

17 Configuring for Redundancy

17.1 Standard CiTect redundancy

This driver is designed to emulate the traditional S7NT driver and it also emulates the error & UNIT status modes of the S7NT driver. This means that CiTect IO device level redundancy failovers work in the traditional CiTect way. Normally this is simply twin CiTect IO servers with the same IO devices configured on both using the same IO device number. The backend configurations are identical. Normally for load sharing the PLC's are divided in two groups, one group primary on server A & Standby on Server B. Primary & standby devices vice versa on IO server B.

17.1.1 Notes

- The Backend status of the pollblocks is 🌿 meaning CiTect is not currently using the data, the backend is actually polling the data every 60 seconds

- **Differences to standard CiTect Drivers**, the main difference is when switching over, the pollblock is in the 🌿 mode on the standby IO server, the every first scan of CiTect data may contain data up to 60 seconds old until the driver polls the new data (<1 second typically).

- **Redundant load balance with only one PLC ?,** If you have a configuration with 1 BIG PLC and a redundant pair of SCADA IO Servers you may run into speed problems. For this reason you may want to load balance the communications between the servers, to do this - setup as follows:

    o Create one backend IO Device called PLC1, both IO Servers use the same
    o Create Two (2) CiTect IO devices, call them something like PLC1_P & PLC1_S
    o Set PLC1_P to primary on IO Server 1 & PLC1_S to Standby. Vice versa on IO Server 2
    o Set both IO Device Address fields to "S7;Ethernet;N=PLC1" the sets the backend link to the N=PLC1
    o Split the tag database in 2 ½ using PLC1_P & ½ using PLC1_S

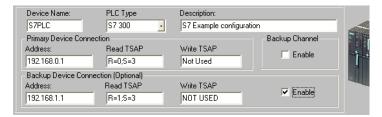### 17.2 Using Backend redundancy with S7-400H systems OR >1 CP card in the PLC

The driver at the backend level has a backup address per IO Device, typically on a S7-400H system you setup as follows: *(***Note redundancy at the backend level means a single Citect level IO device*)

Primary address:

IP 192.168.0.1   Read TSAP = "R=0;S=3"

Backup address:

IP 192.168.1.1   Read TSAP = "R=1;S=3"   (check enable backup address)

*Note** the Rack number of the backup CPU is "1", note I specifically gave the IP addresses different subnets (Ethernet cards in the IO server) you probably have twin Ethernet networks.*

17.2.1 Notes

- A potential issue is that the CiTect Frontend IO device takes 2x longer to failover while the Backend is doing its failover sequence. The timers that affect this are the settings in the channel setup (backend) We recommend settings (do not set any lower)

    o timeout time = 3 sec

    o Delay = 30 sec

    o Retries = 1

- You can monitor which connection the backend is using via the debug addresses

- Unfortunately you cannot at present force the backend to switch over manually

- Read more about monitoring the status in <u>Using the Special Debug addresses</u>

18   Using the S7 Simulator application

18.1 What is it?

The small console application "S7simulator.exe" may be downloaded from a link on

http://www.proscada.com/Citect.htm

18.2 Introduction - why a PLC simulator?

We originally wrote this as a simple application to regression test the driver particularly to test hard to find PLC's like old S5 models.   We soon realised that others would also find it useful for quick test & setup without having to have to setup a *real* PLC.   As such it lowers the barriers to using the PSDIRECT Driver.   We would not want to pretend that it is a "Full" emulation in particular it has not been tested against Siemens software like the OPC server or WinCC. It certainly does not emulate the protocol fully but is sufficient for the ProScada Siemens Driver. It is a low priority project for us, with minimal support.

18.3 Capabilities

- Emulates Both S5 & S7 PLC's (TI sort of…)

- Only Supports TCP/IP

- Automatically emulates any data block you poll with the DB's initialised to []0, any data written is stored, like a PLC with no program running. Emulates 4kb DB's up to DB 1999. M,I,Q memory areas are emulated as DB0

- Supports 2 incoming connections so you can setup a redundant pair of SCADA IO Servers.

- Only Supports Rack 0 Slot 2 (deliberate for training purposes)
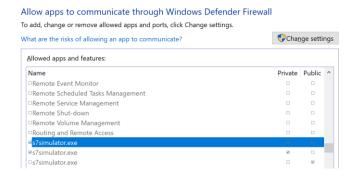
- DB 1 & DB15 contain some forced values

18.4 Installation

Run from the command line or explorer there are no parameters.

When run on the same PC as the IO Server you can use the IP address 127.0.0.1(local loop back) in the driver

The program simply listens for Incoming connections. Being a TCP/IP server you need to unblock any firewall systems. *Note it is receiving connections on TCP port 102.

(network->windows firewall->Allow an app thru)

18.5 Forced variables (internal tags in the simulator with changing values)

DB15 is in S5 format
DB15:0 increments up infinitely every 10msec
DB15:1 increments to pi * 1000
DB15:2 is sin(DB15:1)

DB1 is in S7 format
DB1,2 increments up to 100 every 10msec
DB1,0 increments every time DB1,2 gets to 100 approx. every second

STRING SDB1,128[32] contains "Hello there"

Pseudo code for S7

```
IF DB1,4.0=1 THEN
    DB1,5.6 =1
ELSE
    DB1,5.6 =0
END

IF DB1,4.1 AND DB1,4.2 THEN
    DB1,5.7 =1
ELSE
    DB1,5.7 =0
END
```

DB1,4.3 blink every 1 sec.

Pseudo Code for S5

```
IF DB15,4.0=1 THEN
    DB15,4.14 =1
ELSE
    DB15,4.14 =0
END

IF DB15,4.1 AND DB1,4.2 THEN
    DB15,4.15 =1
ELSE
    DB15,4.15 =0
END
```

DB15,4.3 blink every 1 sec.

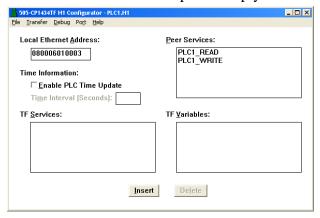# 19   Setting up the Simatic TI

The PSDIRECT Driver replaces the TINECNT & TINECWIN Drivers. The TINEC drivers use the Siemens TF interface via the CP1413 card. The ProScada Ethernet backend can talk to the PLC via ISO or TCP/IP protocols, however note that it does NOT support TF. The Siemens TF system is little more than an unnecessary abstraction layer and in reality it is quite easy to transfer the system.

The PSDIRECT Installation installs a example project in "Citect\bin\PSDIRECT samples"
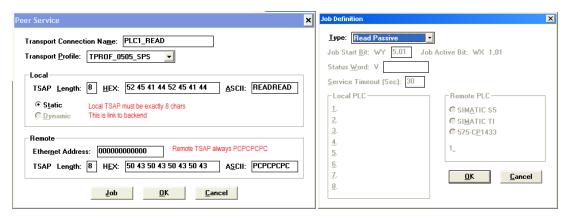
The steps to migrating an existing system are as follows:

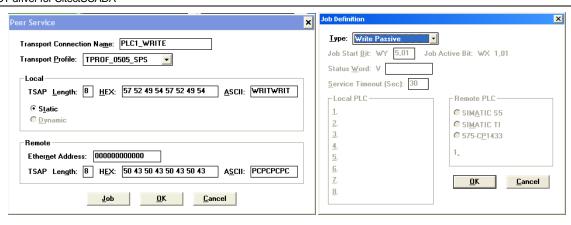### 19.1.1 Step 1: Adding peer-peer jobs to the PLC CP card setup.

You can leave the old TF Jobs in place simply add these new "Peer Services".

They look like this

- Local TSAP must be exactly 8 chars

- Local TSAP is the link to the backend configuration see below

- Remote Ethernet address should be "0000" – accepts any incoming connection

- This should be a read Passive job

- The Write job should be a Write passive

- All jobs remote TSAP must be "PCPCPCPC"

19.1.2 Configuring Redundant SCADA servers

To allow multiple SCADA's to access the PLC you need to configure a pair of the above "jobs" per SCADA IO server. In this case it is recommended to name the Local TSAP linking to the IO server name remembering it MUST be 8 chars. e.g. IOSPRIMR, IOSPRIMW & IOSBACKR, IOSBACKW

19.1.3 Step 2: Configure the backend driver

- The TCP/IP version of the CP1434 card is quite rare, so most will use the ISO transport (MAC address) not TCP/IP. If using TCP/IP, do not proceed until you can ping the PLC.

  o Setup the Siemens ISO transport in the network settings, this is the same as for the S7 & S5 PLC's described earlier in this document

- Set the read & write TSAP's to the Local TSAP names in the CP1434 config

- Check that you have communications, you should do this before setting up CiTect. Simply setup 1 block of "V" registers. You should see green ticks & blocks per second at the bottom of the configurator window indicating successful communication

19.1.4 Step 3: Configure the backend blocks

- Set the name of the backend IO device the same as the IO device in CiTect

- Open the project variable.dbf file in excel, filter on IO device & sort on IO Address. Then manually add equivalent blocks in the backend. ****Note the backend addressing is slightly different, it has ":" between blocktype & address eg. V1000->V:1000

- At present there are no Learn address & variable import features for TI.

- Any Citect tag accessing data in a block not being polled, will cause "Nio-*Address error – xxx" messages. You **MUST** get rid of these errors by adding poll blocks

- **Note Changes to the back end only activate after File>Reload

19.1.5 STEP 4 : Configure Citect

- Remove all TINECNT ports, Just add 1 port/channel for PSDIRECT (I use PSPORT)

- In the IO Device fields

  o Set PROTOCOL to PSDIRECT4

  o Set address to "TI"

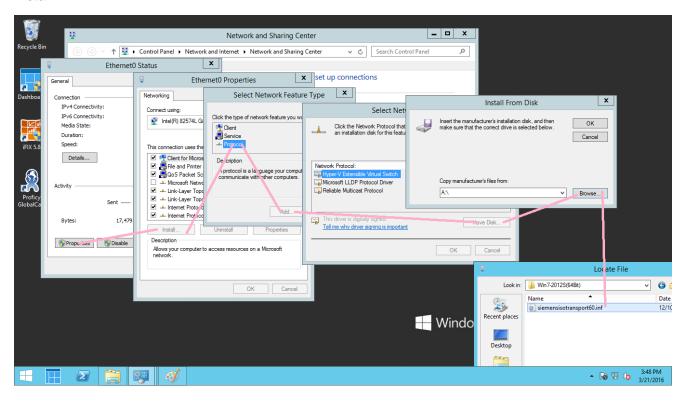 o Set ALL IO devices port to the common port for PSDIRECT (PSPORT)

20   Installing "Siemens H1 ISO transport" (***Not used for TCP/IP)

*This step is **optional**, depending on the protocols used to talk to the PLC's, if you are using TCP/IP you DO NOT Require the ISO transport – Please go to Section 4 Setting up the driver.
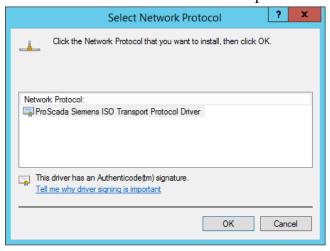
The ISO transport is an older Siemens protocol that requires a 64bit Kernel mode Windows driver called "ProScada Siemens ISO Transport".
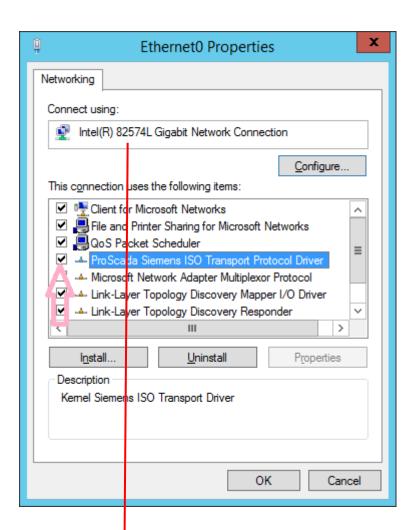
To Install:

1. Copy the contents of the zip file directory Win 7 – 2012S (64 bit) to a user directory.

    Open Network and Sharing centre > Change Adapter settings > R click properties and click Install
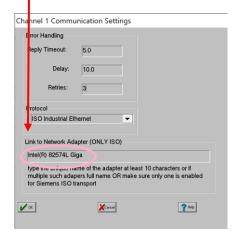

2. Select the ProScada Siemens ISO transport

3. Now open properties on ALL other network adapters and unclick the "ProScada Siemens ISO transport"
4. Uncheck all Internet Protocol Version x (TCP/IP) & all Microsoft networking leaving only ISO on this ethernet card (*note you will need >1 ethernet adapter)

5. Now go to the Front end (SIXDIDW.EXE) driver configurator and press setup button on the channel settings type at least 11 characters of the ethernet card name in "Link to Network adapter"

6. Now reboot the machine, run the driver configurator "AS Administrator" & check that you get the message ISO Channel opened on adapter x address nn:nn:nn….
   a. If you get "unable to open SC" it means you are not running as admin
   b. It may also fail to connect if the ethernet cable is unplugged
   c. The Siemens ISO transport, should run at startup after a reboot but if it does not, you need to shut down the driver and try again (AS Admin)
   d. To debug the IOS transport service open a CMD Box with admin rights and type
      i. "NET START SIEMENSISOTRANSPORT<enter>
      ii. It should say Already started or successfully started

21   Using the Square D PLC Backend with CiTect

The PSDIRECT Frontend emulates the "SQUARED" PLC driver

21.1 Installation

1.   Install the PSDIRECT front end

2.   Install the Square D low level protocol as described in the SDEDIDW.HLP help document done in Control panel / networking/

3.   Enable this protocol only on the network adapter you are going to use, Disable on all other wireless, dialup, VPN, VMW adapters on the PC

4.   Reboot

5.   Install the Square D Backend driver (setup. Exe) – This may crash – just continue

6.   Run the driver configurator part may crash,

7.   Look in the bottom window if there is a message like unable to open Square D adapter that is causing the crash. Read in the help file how to fix in the registry.

8.   If it successfully opened the correct Ethernet card it will not crash again.

9.   Setup a device with correct Ethernet address & add a chunk to poll

10.   Whenever you make changes here you have to "File>reload" yellow pc button

11.   When talking you get green tick marks & polls per second bottom left

12.   Now setup CiTect start with the example project

21.2 CiTect channel Form

Create one channel object, leave the Boards field blank

21.3 CiTect I/O Devices Form

Set the Protocol Field to "PSDIRECT7" set all IO devices to use the same channel(only 1) object

21.4 Data types

Data Types	Address Format	CitectSCADA Data Type
BITS	*w - b*	DIGITAL
REGISTER	S*w*	BCD / INT / LONG / LONGBCD / REAL

Where

w is the word number 1 to 32767 (the maximum is actually determined by memory constraints of the I/O Device)

b is the bit number 1 to 16

EXAMPLES:

Data Type	DIGITAL
Address	01-15
Comment	Word Number 1 - Bit Number 15

Data Type	INT
Address	S0001
Comment	Register - Number 1

SY/MAX PLCs support remapping reads and writes.

Using the IEC 870 Backend with CiTect

The PSDIRECT Frontend emulates the addresses of "IEC" driver

For further information this should be read in conjunction with :

http://proscada.com/DOC/Interoperability Definition for IEC 870-5-101&104 Protocol.pdf

21.5 Note on installation
- Install the PSDIRECT front end and the IEC 870 backend
- Ensure there is a copy of the file I87STSR.DLL in the Citect\Bin directory

21.6 CiTect configuration

21.6.1 Boards Form:
None

21.6.2 Ports / Channels Form:
Same as other PSDIRECT profiles just create one port with port number 0

21.6.3 IO Devices / Units Form:
All IO Devices can link the one port
The name links to the backend RTU name
The Protocol is "PSDIRECT5"
The address is the IEC "Common asdu address" of all the tags on this IO Device, 0-65536

21.6.4 CITECT.INI Settings

To configure Time Stamped Alarms you need to set the following option because the Timer Expr. Value for this driver is a 24h Millisecond counter.

[Alarm]

HresType=7

[Alarm]

HighResOff=1   ; Usually set to rising & falling edge

[Alarm]

Hres24HrDeadBand=3600   ;set to 1 hour

21.7 Input tag values

**Note for DPI & SPI the driver data represents a 16 bit INT this can be accessed either by bits 0-15 in DIGITAL tags or as an analogue with INT tags – Same memory

The bits are accessed by string names as follows (note the '.' separator)

SPI & DPI
VAL=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7,SELECT=7,SPUL=2,LPUL=3,NEGC=14,ISSELECTED=15
QMV & QIT& QFLOAT & QBITS
<OV=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7>

	CiTect Tag Addressing Notation	CiTect Data Type	Comment
SPI Value	SPI:<Object No>.VAL	DIGITAL,INT	VAL=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7,SELECT=7,SPUL=2,LPUL=3,NEGC=14,ISSELECTED=15
SPI Invalid	SPI:<Object No>.INVALID	DIGITAL,INT	VAL=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7,SELECT=7,SPUL=2,LPUL=3,NEGC=14,ISSELECTED=15
SPI Time	TSPI:<Object No>	LONG	Represents millisec after midnight
DPI Value BIT 0:	DPI:<Object No>.VAL	DIGITAL	VAL=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7,SELECT=7,SPUL=2,LPUL=3,NEGC=14,ISSELECTED=15
BIT 1:	DPI:<Object No>.1	DIGITAL	VAL=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7,SELECT=7,SPUL=2,LPUL=3,NEGC=14,ISSELECTED=15
DPI Invalid	DPI:<Object No>.INVALID	DIGITAL	VAL=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7,SELECT=7,SPUL=2,LPUL=3,NEGC=14,ISSELECTED=15
DPI Time	TDPI:<Object No>	LONG	Represents millisec after midnight
DPI Value as INT	DPI:<Object No>	INT	then use Cicode functions to strip & mask out values e.g. 0-1 DPI as analogue
MV Value	MV:<Object No>	INT	** contains both scaled & normalised data – whatever received
MV Invalid	QMV:<Object No>.INVALID	DIGITAL	<OV=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7>
MV Time	TMV:<Object No>	LONG	Represents millisec after midnight
MV Float Value	FLOAT:<Object No>	REAL	
MV Float Invalid	QFLOAT:<Object No>.INVALID	DIGITAL	<OV=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7>
MV Float Time	TFLOAT:<Object No>	LONG	Represents millisec after midnight
Bitstring Bit Value	BITS:<Object No>.<bit>	DIGITAL	Bits 0-31 as decimal
Bitstring Invalid	QBITS:<Object No>. INVALID	DIGITAL	<OV=0,1=1,2=2,3=3,BLOCK=4,SUB=5,TOP=6,INVALID=7>
Bitstring Time	QBITS:<Object No>	LONG	Represents millisec after midnight
IT Value	IT:<Object No>	LONG	Counter value, write to bits 0-7 sends those bits to counter e.g. reset
IT Invalid	QIT:<Object No>.INVALID	DIGITAL	CY=5,CA=6,INVALID=7
IT Time	TIT:<Object No>	LONG	Represents millisec after midnight
STP Value	MV:<Object No>[.TRANSIENT]	INT,[DIGITAL]	Bits 0-6 step value
STP Invalid	QMV:<Object No>.INVALID	DIGITAL	0=0,1=1,2=2,3=3,4=4,5=5,6=6,TRANSIENT=7
STP Time	TMV:<Object No>	LONG	Represents millisec after midnight

21.8 Status data tags

	Tag Addressing Notation	Data Type	Comment
Master RTU STATUS	STATUS:0	INT	0=Successful master polling, 1= RTU failure
Master RTU reset comms	STATUS:0	INT	Write 1
Slave RTU is polled timer	STATUS:2	INT	Decrementing timer, reset up on every masterpoll – 0=master not polling
FORCEGI	FORCEGI:<Object No>	INT	Triggers interrogate, write value is the Interrogate group 20 = all

21.9 Tag Setpoints & Controls

	CiTect Tag Addressing Notation	CiTect Data Type	Comment
SPI Select ON (Long Pulse)	SPI:<Object No>	INT	Write 137
SPI Select OFF (Long Pulse)	SPI:<Object No>	INT	Write 136
SPI Execute ON (Long Pulse)	SPI:<Object No>	INT	Write 9
SPI Execute OFF (Long Pulse)	SPI:<Object No>	INT	Write 8
SPI Selected Feedback	SPI:<Object No>.SELECTED	DIGITAL	Indicates after RTU has replied to Select command
SPI Negative Acknowledgement feedback	SPI:<Object No>.NEGC	DIGITAL	Indicates after RTU has replied to Select command
DPI Select ON (Long Pulse)	DPI:<Object No>	INT	Write 138
DPI Select OFF (Long Pulse)	DPI:<Object No>	INT	Write 137
DPI Execute ON (Long Pulse)	DPI:<Object No>	INT	Write 10
DPI Execute OFF (Long Pulse)	DPI:<Object No>	INT	Write 9
DPI Selected (Feedback)	DPI:<Object No>.SELECTED	DIGITAL	Indicates after RTU has replied to Select command
DPI Negative Acknowledgement feedback	DPI:<Object No>.NEGC	DIGITAL	Indicates after RTU has replied to Select command

21.10 Time Stamped Alarms

This Driver supports RTU generated timestamps, to use this feature use Configure Time-stamped Alarms

Configure the Variable Tag – typically this is a Digital addressed to the VAL bit e.g. SPI:<n>:VAL for analogs you will need to define your alarm in a Cicode function that returns a Digital.

The Timer Expr. Field MUST be set to a LONG tag addressed to one of the "T" addresses e.g. TSPI:<n>

Where <n> is the same in both Fields

**Note this CITECT.INI setting

[Alarm]
HresType=7