



FET | VisualSoft

REMOTE OPERATION OF SUBSEA VIDEO RECORDING

Bill Parker-Jervis, VisualSoft Technical Manager

Company Introduction

FET

- Created as **Forum Energy Technologies** in 2010 through a merger of multiple companies
- Headquarters in Houston, Texas
- Products and infrastructure for onshore and offshore oil, gas and renewables
- Well known legacy brands across various divisions

www.f-e-t.com



Subsea Division

- ROV Manufacturing
- ROV Tooling & Component Manufacturing
- ROV Tooling & Component Rental
- Video and Data Acquisition & Processing
- ... and more...

VisualSoft

- A leading player in subsea digital video and data acquisition, processing, display and reporting tools

FET was formed in 2010 through the merger of various companies, and it has a number of different divisions serving different parts of the market. The Subsea Division includes ROV manufacturing, ROV tooling and component manufacturing and now also rental of that equipment. There is a lot more to the Subsea Division and to FET as a whole and you can find details of that and our many legacy brands on our website.

VisualSoft itself was founded as a company with a couple of people 25 years ago, and now sits within FET's Subsea Division. The software and hardware produced by the VisualSoft team, makes up the VisualSoft Suite which has digital video at its heart and also handles other data types for surveys relating to construction and inspection. The use of our video recording application, VisualDVR, will be the focus of this presentation.

VisualDVR and Remote Operation

- How is VisualDVR used now?
- What do our customers want and why?
- What has been tried and how did it work?
- What are we doing to improve on that?
- How will the remote-control application look and feel?



What I plan to talk about today will cover how our systems are currently being used offshore, how people have experimented with doing things differently, and how we aim to improve the software to allow for remote operation and remote control.

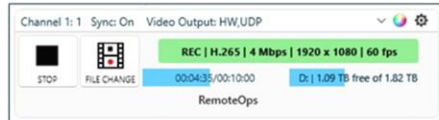
In the last few years many companies have been very focussed on improving safety and reducing costs. The days of huge budgets are gone, and the awareness of safety has increased, so we all need better ways of operating. One way to achieve these goals of better safety and lower costs is to reduce the number of people being sent offshore.

A lot of work is being put into autonomous surface vessels and autonomous subsea vehicles. Efforts are also being made to remotely operate systems on current, non-autonomous ships and ROVs. If we can't, or can't yet, have fully autonomous systems then a good starting point is to begin reducing the manning levels of offshore crews, and we can assist that by making it possible to operate things remotely.

FET is one of many companies actively working on Remote Operations and we're doing so on multiple fronts. This includes remote operation of ROVs from onshore bases and also remote operation of the survey and inspection equipment carried by and connected to those vehicles... which brings us to VisualSoft's work on the remote operation of VisualDVR.

VisualDVR now

BILL-44235VisualDVR_Ch1 "Recording"				BILL-44235VisualDVR_Ch2 "Recording"			
Time(UTC)	Time(Local)	Computer	Application	Message			
13:29:21	13:29:21	BILL-44235	VisualDVR	VisualSoft File Service is not running. Starting now.			
13:29:20	13:29:20	BILL-44235	VisualDVR_Ch2	File renamed and ready for transfer "E:\VisualWorks\Encoder\BILL-44235_Ch2\20240125132538876@BILL-44235			
13:29:20	13:29:20	BILL-44235	VisualDVR_Ch3	File renamed and ready for transfer "F:\VisualWorks\Encoder\BILL-44235_Ch3\20240125132538876@BILL-44235			
Date	Time	Fix	Easting	Northing	KP		
25/01/2024	13:30:00	5600	502808.095	6702800.422	3.964121		87



Although the capabilities and quality of video systems for subsea survey have changed over the last 25 years, quite a lot of how they are operated hasn't changed. We still need someone able to see the DVR to check that cameras are pointing the right way, the view is adequately lit, and the recorder is doing what it should, and that person may also need to log inspection events in real-time.

VisualDVR operation

- User in front of the DVR PC



- User elsewhere on a ship



- User onshore in an office



How all of that is done might be one of the following:

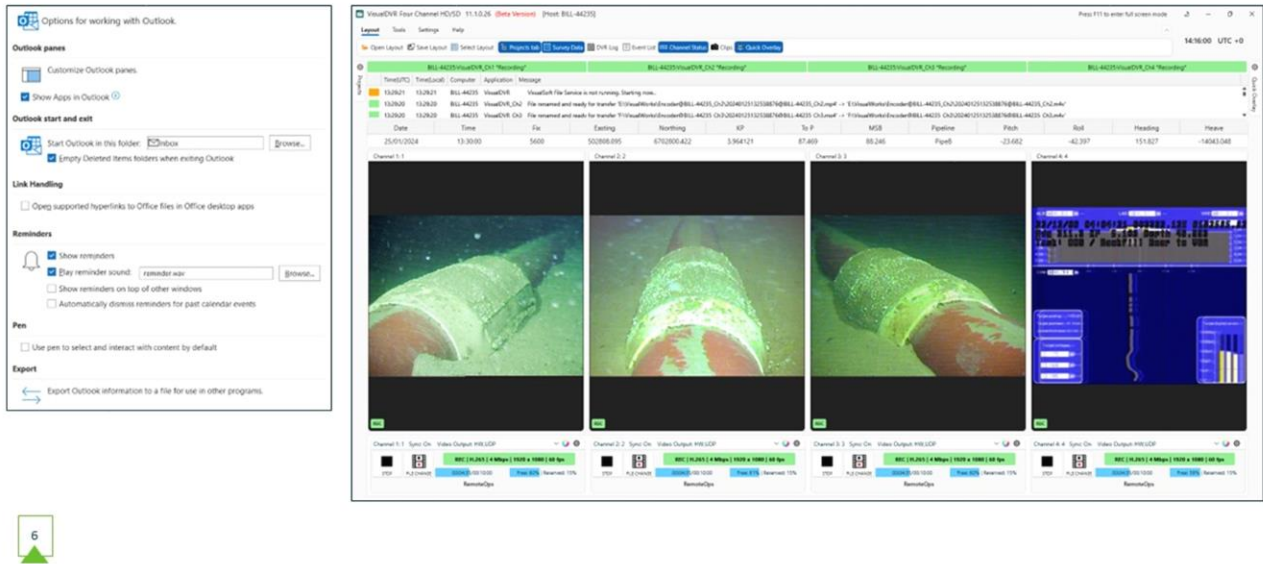
- Someone sitting right in front of the DVR PC,
- Someone elsewhere on the same ship, using remote access software of some sort,
- Someone onshore using remote access software of some sort,
- ... and what we're working on now, which is someone onshore using a bespoke remote viewer and controller.

For many years some of our customers on larger ships have been using remote access software such as Remote Desktop, or TeamViewer to allow people in a Survey control room to view and operate a DVR that's in an ROV control room on the same ship. The network bandwidth in this scenario is enough that the live video display can be shared across the ship's internal network with negligible delay or loss of quality. For some companies this way of working is now the norm as it removed the need for long coax cable runs. They can leave the DVR hardware close to the source of the camera signals coming up from the ROV but operate that DVR from elsewhere on the ship. Where both PCs are on the same internal network on the same ship this has generally been quite successful.

In the last few years, some of our customers have also been experimenting with controlling their DVRs from offices onshore, and in this case the results have been less than perfect.

Remote access applications like TeamViewer or Microsoft's Remote Desktop are largely designed for troubleshooting IT problems, and so they're intended to allow things to be looked at and re-configured, with the connection to the remote PC only lasting a short time.

Remote Access, and Display Complexity



In the scenario of IT troubleshooting, the display being accessed is largely static – a dialog box with some text, some buttons, and some checkboxes like the image on the left of this slide. A good remote access application doesn't need to update the whole display all the time and can just update what changes. If the change is simply a checkbox changing from "selected" to "not selected" then that's a tiny part of the screen that needs to change. Even if the whole display does need to update it will typically have large areas of solid colour that compressed very effectively.

The DVR display on the right is very different from that, with video previews changing at a refresh rate of 25 to 60 times per second and it's not just small areas, or blocks of solid colour, that are changing, but just about every single pixel changing colour with every refresh.

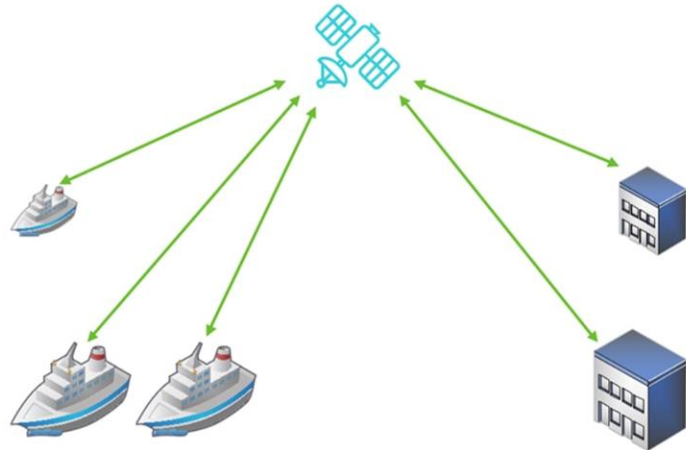
There's other data such as positions, depths, KPs, compass headings and so on that's also updating rapidly.

Remote access software will generally try to update the local display as soon as the remote one changes, so that as you click on buttons and checkboxes it feels quick and responsive. That's great for a typical IT support use case, but when the software tries to keep up with the ever-changing megapixels of video, it soon ceases to feel responsive and can become quite unworkable.

Bandwidth and Cost

- Bandwidth rising
- Costs falling
- Requires flexible software design...

Something designed to be useable now, but get even better later



Both the latency and the cost of streaming video from ship to shore will significantly reduce as Starlink and similar systems become more widely used, and so streaming multiple video channels at high resolution, high framerates, and high quality will in time become the norm.

However, we are seeing the demand for remote operation of our DVR systems coming ahead of the change to affordable, high bandwidth streaming, and so we need to devise a system that will work now and will also be scalable and adaptable to ramp up performance for the better-bandwidth days ahead.

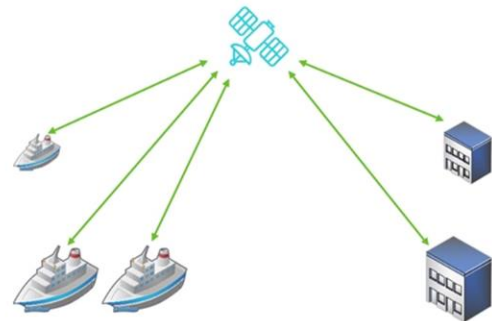
We also need to allow for recording at either end – offshore or onshore – because as latency falls and quality doesn't need to be reduced for streaming, then it may make sense for the recording process to move from the ship to the office, or more likely, we'll probably need to record in both places at once so that nothing is lost when a comms link drops out. If the offshore DVR keeps going regardless, then users could plug any gaps in the onshore version of their data once the comms link is restored.

Remote Control: Design Guidelines

- **Retain a familiar look and feel:** If you're operating VisualDVR it should look like VisualDVR
- **Scalable video preview:** Adjust framerates and perhaps resolutions to suit available bandwidth
- **Switchable between local and remote operation**
- **Cope with loss of comms**
- **Scalable "control stations":** Control one DVR or many?

On one ship or many?

From one office or many?



That brings us to what we're working on to make VisualDVR remotely controllable and I want to talk about some of the main design features of that:

We want the operation of VisualDVR to look and feel the same whether you're sitting in front of it, sitting in a different room on the same ship or sitting a thousand miles away in an office onshore. Someone who normally works offshore shouldn't need to learn their way around a whole new application just because this week they're operating a DVR remotely.

We want to scale the refresh rate of remote preview displays to suit the available bandwidth. We need something that works with both low bandwidth and high, altering the update of the video displays to suit what's available. If the connection is slow, we should slow the refresh of the previews so that we don't use all the bandwidth for video at the expense of making the application sluggish and un-responsive.

If bandwidth is low we can look at whether we really need HD previews, or if we should leave the offshore unit in HD mode, but downscale what is sent ashore to be SD. As long as the previews are enough to show users that their cameras are viewing what they should then let's not make everything feel sluggish for the sake of a faster refresh or full-HD image. But equally – if and when we can have fast, full resolution video and still have everything else responsive and quick then we should.

Not every remote operated DVR will be on an uncrewed vessel. There will be times when remote control is used to assist or to train people or to troubleshoot and in these cases we need to be easily able to switch a DVR between local operation and remote operation, and then switch back again.

We want a DVR to carry on doing what it should even if the comms link over which it was being controlled is lost. A comms failure shouldn't mean that recording stops and shouldn't mean that someone local to the DVR can't flip a switch and take over if necessary.

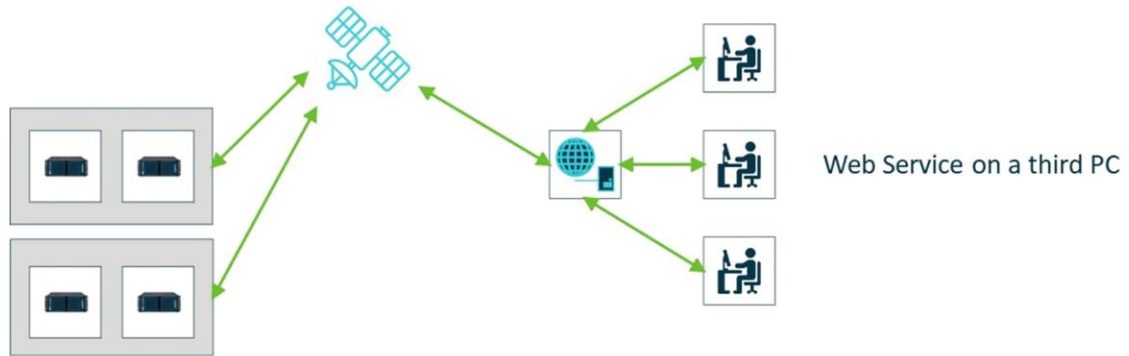
We also want to provide a scalable "control station" that can manage just one DVR or can manage multiple DVRs on multiple ships.

Remote Control: Comms



We are using a Web Service for the comms between the remote-control application and the DVR that it's controlling. This fits well with that need for scalability. The host for the web service can be the same onshore PC where the remote-control software is being run or it can be the offshore DVR, or it can be a third PC, perhaps even in a third location. This allows flexibility around how different things sit within a company's firewalls, switches and other security measures.

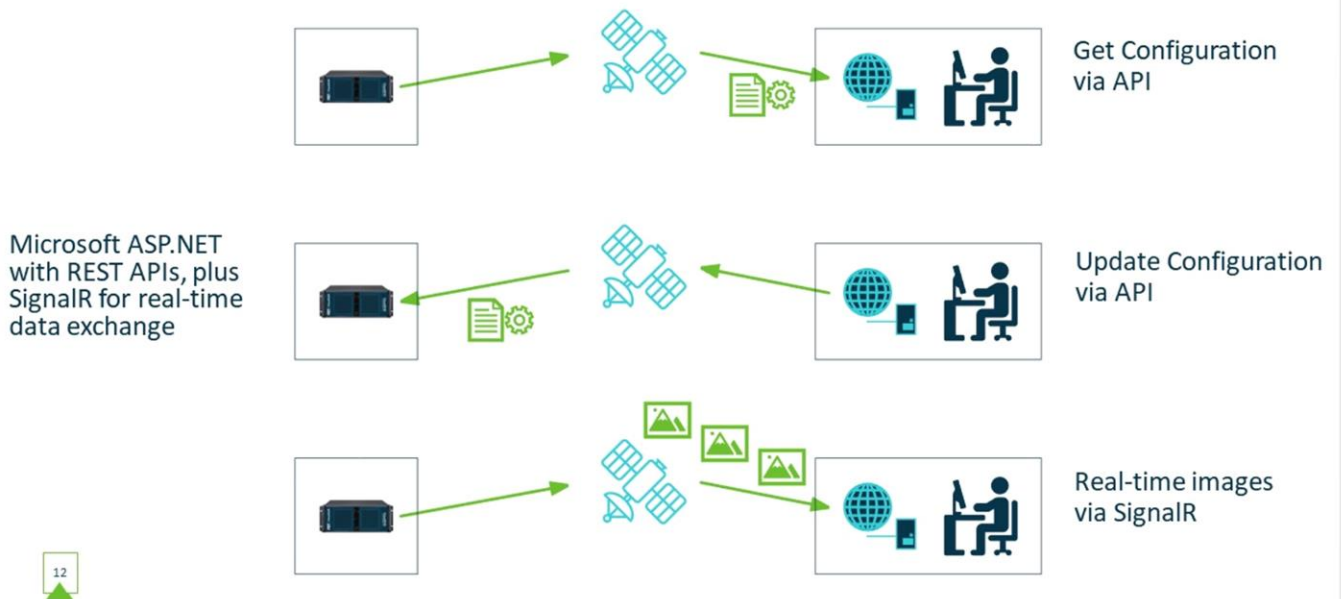
Hosting and Scaling the Web Service



11

For smaller and simple setups, we would expect the Web Service to be most often run on the same PC as the remote-controller, but for customers wanting access to multiple DVRs and multiple ships it would make more sense for the Web Service to run on a third machine, and potentially one that they are using for other intranet functions so that the necessary routing and firewalls between ships and offices is already largely in place. As shown here we can scale from one-to-one connections up to one-to-many.

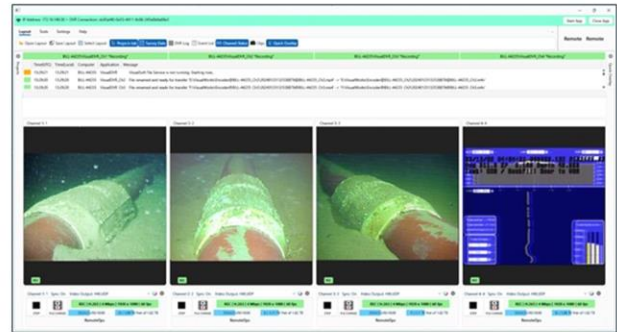
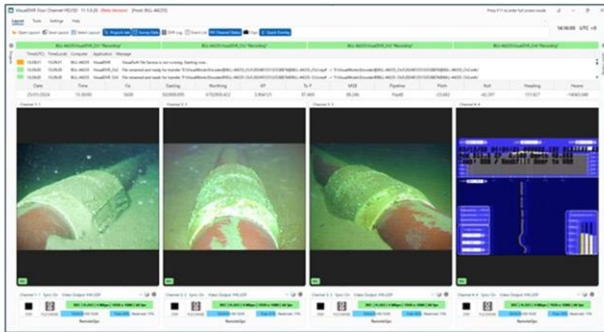
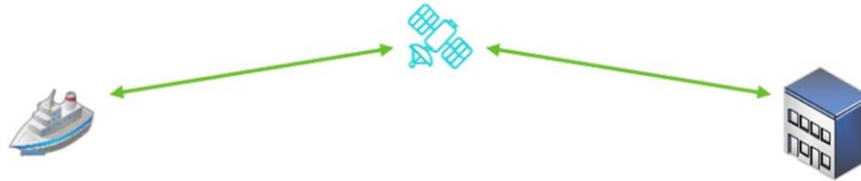
Basics of the API and comms



Communication between the various PCs uses Microsoft's ASP.NET WebAPIs and also SignalR, which makes it relatively easy for us to send and receive configuration information from PC to PC in real-time. We can send a REST API request to GET the configuration of the offshore PC and use that config to make an application on the onshore PC look the same as the offshore equivalent. The onshore user can then change any settings locally, and we use the APIs to send the new configuration back offshore and apply it just the same as if someone sitting at the DVR had changed them.

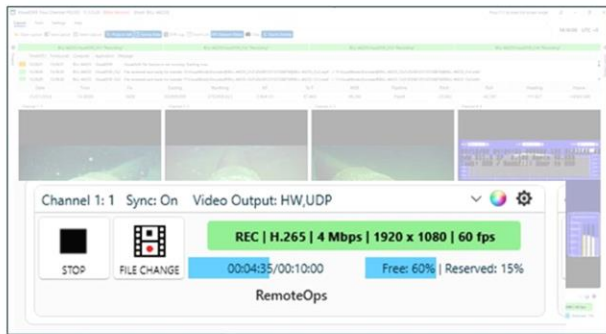
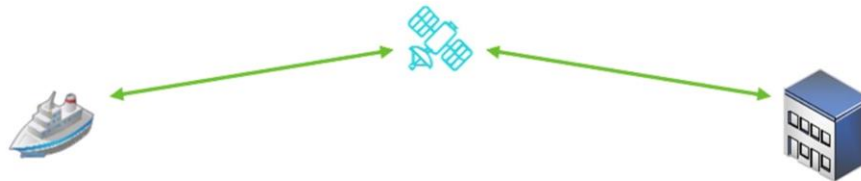
Similarly, we can have a sequence of images from the offshore DVR, sent "instantly" by SignalR, so that we get an almost live preview on the instance in the office.

Offshore and Onshore Displays



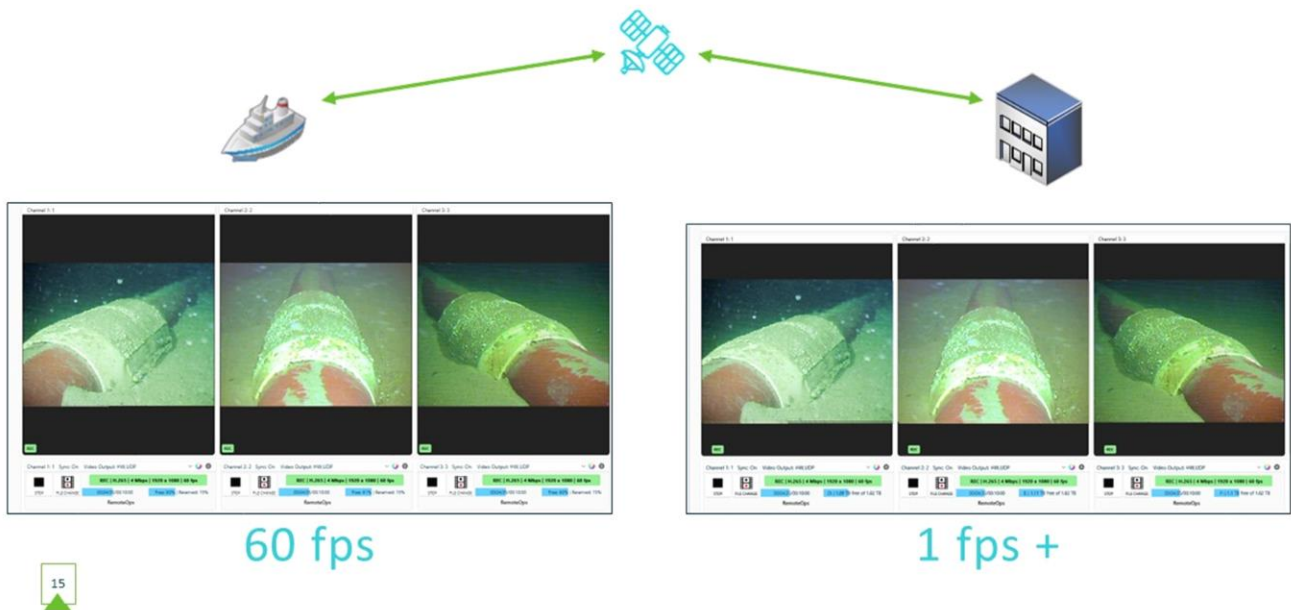
As I mentioned earlier, we very much want the onshore user to see and interact with the DVR in the same way that they would if sitting in front of it offshore – or as close to that experience as we can get with the available comms bandwidth. Therefore, our Web Service doesn't only need to update settings and receive preview images, but we also need the underlying data that drives things like status displays, warnings, and alarms. For example, we're sending information about disk-usage, logging status, and recording duration so that the onshore screen matches the offshore one.

Offshore and Onshore Displays



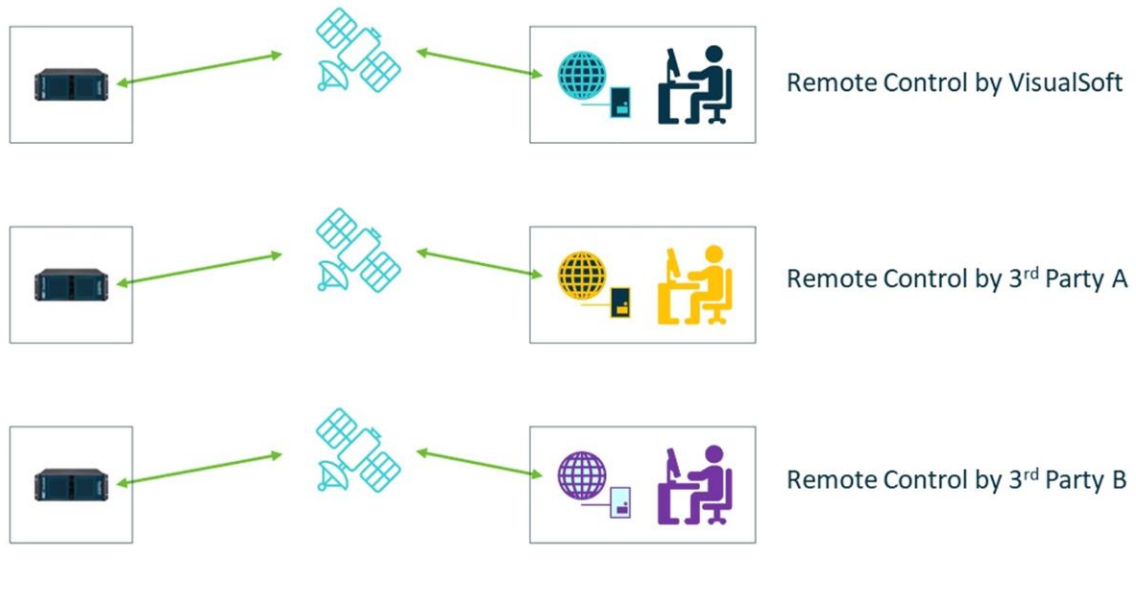
For those status displays at the bottom of the screen, all that is being sent from the ship is a few numbers and some text strings, and we're using that data in what is essentially the same application onshore and offshore to display the information.

Offshore and Onshore Displays



The same applies for the video: Offshore we have preview windows updating 60 times each second from the live camera feed, but the same application running in the onshore mode might be getting, perhaps, one compressed image each second, so we have something that looks basically the same in each location but with a huge difference in how much data the applications are using to create the two displays.

As the bandwidth from ship to shore improves, we can ramp up the onshore refresh rate until we get to a point where what we're seeing onshore is no longer a sequence of still images but instead becomes the full video stream.

3rd Party use of the new API

A side effect of what we're building for the Web Service will be a reworking and updating of our current API to make that more powerful and more flexible. If we're making it easy for ourselves to remotely control our DVRs then it's an ideal chance to also improve on how 3rd party systems might also remotely control those DVRs. For about 20 years we've had an API that allows various inspection packages and other applications to control parts of our system, and the development of our new remote-operations tools gives us an ideal chance to review that API and make it faster and better. The use of a Web Service for the next generation of our API ties in well with what others are doing around cloud-based inspection systems.

Security of Remote Operations



-  Security tokens built into VisualSoft applications
-  HTTPS, and IP restrictions
-  User configurable keys to define groups of PCs
-  ... and more... Please get your security teams to talk to us.

It's clear that if we're going to allow onshore and offshore PCs to communicate via intranet or internet, then we're going to need security measures. This isn't the right place to talk about the details of it, but the Web Service implementation handles security in a number of ways.

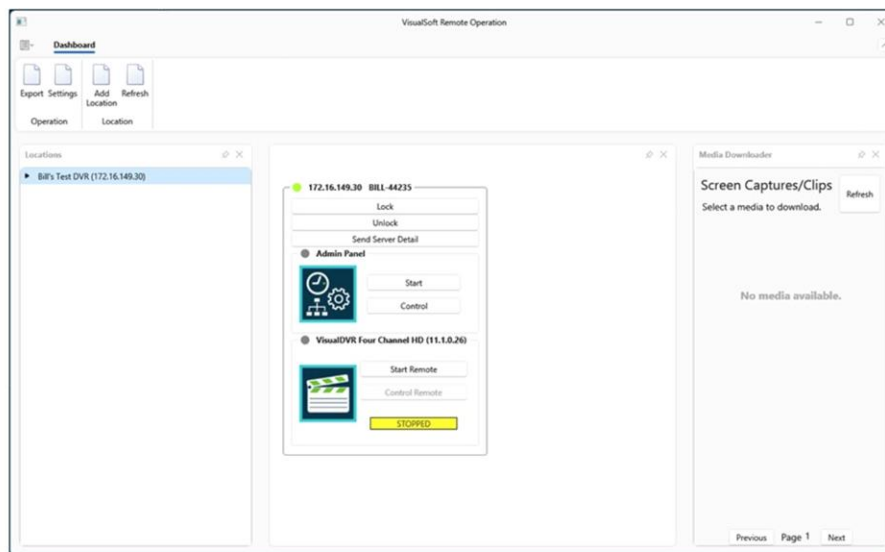
Built into each of our applications are complex security tokens that provide the first level of protection through the fact that unauthorized applications wouldn't match those tokens and any API request that doesn't have the right token will be ignored.

The API Key set on our Web Service for securing the API and SignalR won't be accessible to unauthorized people, and allowing the use of **HTTPS** for the Web Service ensures secure communication by encrypting data transmissions. Other aspects of security include restrictions on IP address ranges and the blocking of unexpected traffic.

For the simpler task of stopping users in an office from accidentally connecting to DVRs on the wrong ship, we also have a simpler, user configurable API key which needs to match on the onshore and offshore PCs. That key could be something as simple as the name of a ship or a project, determining which onshore users can access which offshore systems.

We are very keen to hear from the network security experts employed by our clients – but perhaps not here and now! We're still working on this development so there is plenty of scope to incorporate security recommendations from those companies most likely to use these new applications.

Setting Up for Remote Operations



18

To configure the systems for remote operations, we will have a service running on each of the PCs, onshore and offshore. On offshore systems that service will be configured to start at boot time, and for onshore PCs we can either do the same, or we can start the underlying service when a user starts the Remote Operations application.

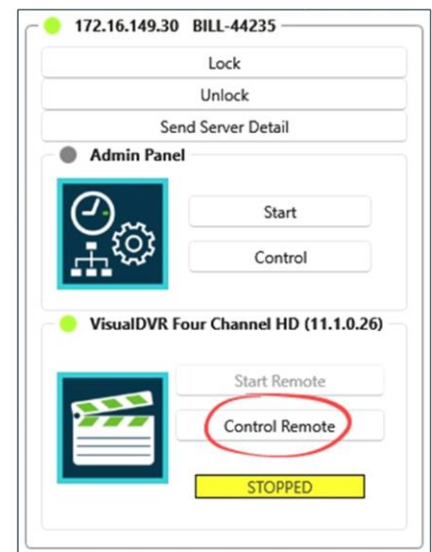
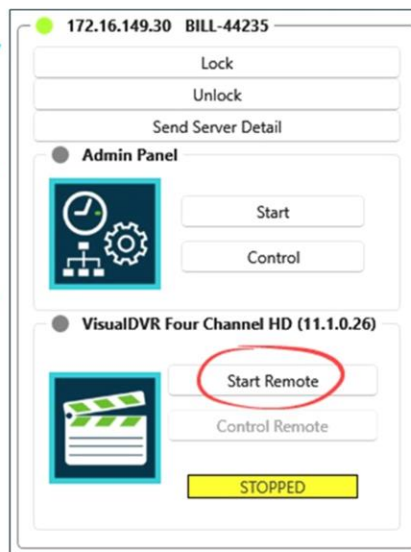
The Web Service too will need to be running on whichever PC or server hosts it, and again we'd configure that to run at boot time.

Once you start the Remote Operations application, you will see a simple dashboard where you can add one or more "Locations". These Locations will most often be individual ships with one or more PCs on board hosting applications that you want to control.

Setting Up for Remote Operations

Connection to remote PC

Status of applications on the remote PC

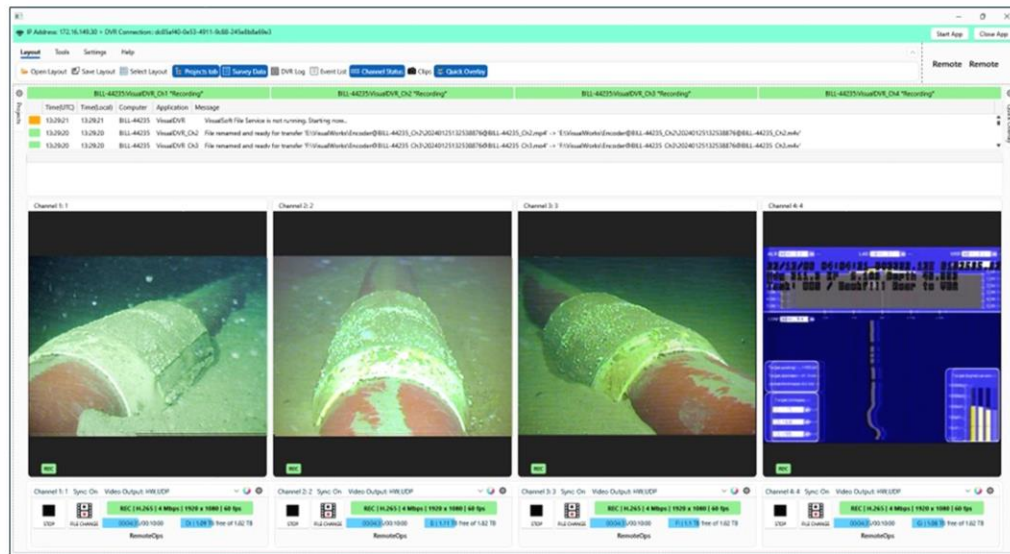


19

For applications that can be remotely controlled there's a standard application icon, matching what's in the Start menu of the offshore PC, and there are buttons to **Start** that remote application, and then once it's started, to take **control** of it.

Clicking on **Control Remote**, will start up a local copy of the DVR application.

Onshore Display



20

Once the local view is running it can then be used to monitor and control the offshore application. So far, we've implemented some of the basic controls, starting and stopping recording and grabbing still images for example, and we're working on adding the API and comms needed for the onshore instance to access all settings and controls on the offshore application.

Clicking the buttons on the onshore instance of the DVR software shows you exactly the same dialog boxes and prompts that you would see if you were offshore; the only difference is that when you do things like select logging disks or choose video inputs those disks and inputs are on the offshore PC not the local one.

What's Next

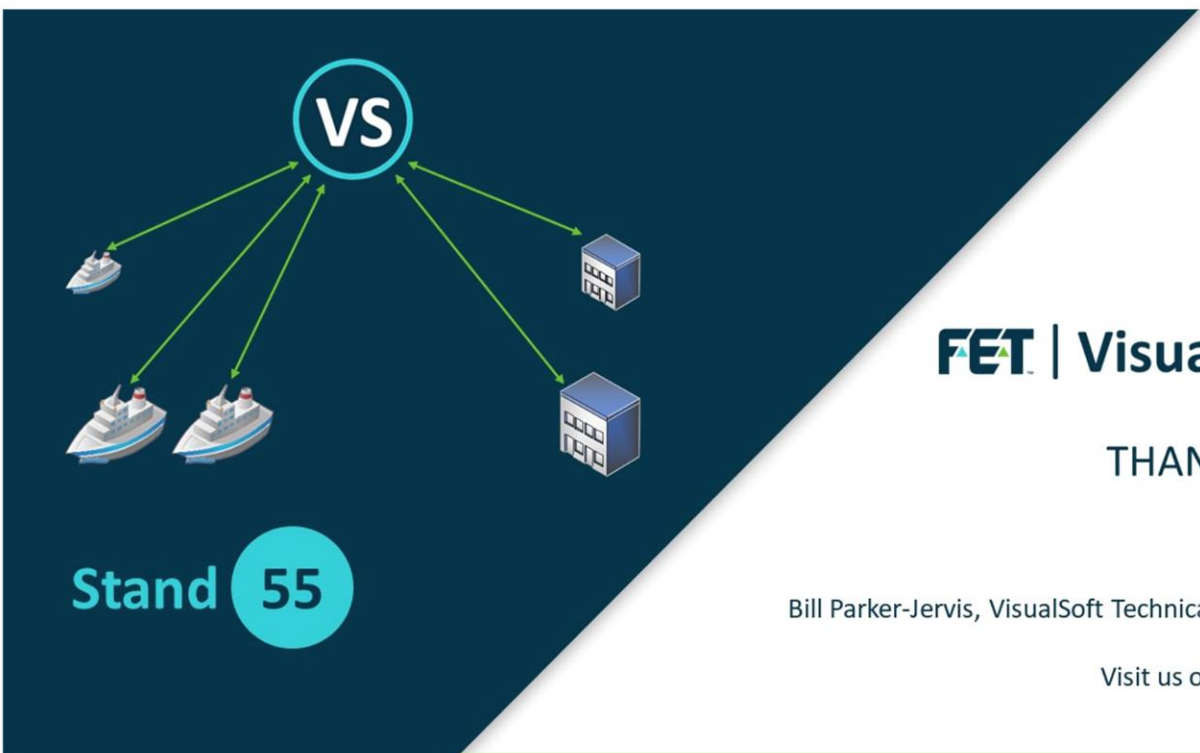
- Finish DVR implementation
- Add Remote Operation to our Admin Panel for licence management and troubleshooting
- Add Remote Operation to other applications such as Event Logger, VisualArchive and VisualData Logger

Demos

- We have live demos of a beta version on **stand 55** here, and next month at Oceanology in London



We still have some work to do to complete this implementation for full Remote Operation of VisualDVR, but once that's done, we aim to add similar remote capabilities to other VisualSoft applications that are typically used alongside our DVRs. That includes the Admin Panel used for licensing, time and network settings, and troubleshooting, then our survey data and event logger applications, and our data management and backups in VisualArchive.



VS

Stand 55

FET | VisualSoft

THANK YOU

Bill Parker-Jervis, VisualSoft Technical Manager

Visit us on Stand 55

If anyone wants to see our Remote Operations tool for themselves, we can demo an early work-in-progress build of it on our stand here this week, or next month at Oceanology in London.