Distributed OrcaFlex

1 Introduction

Distributed OrcaFlex is a suite of programs that enables a collection of networked, OrcaFlex licensed, computers to run OrcaFlex jobs as background tasks using spare processor time.

Distributed OrcaFlex consists of four separate programs:

- A Distributed OrcaFlex **server program** runs on one machine in the network. The server manages the job list, allocates jobs to the client machines, and relays communication between the clients and viewers. The server program does not require a licence.
- A Distributed OrcaFlex **client program** runs on each machine available to run jobs. This program loads the required OrcaFlex DLL version and runs the jobs at the request of the server. The **client program** claims an OrcaFlex licence when running jobs.
- A Distributed OrcaFlex **viewer program** can run on any machine on the same network as the other DOF components, it does not require a licence. The viewer program displays the list of jobs and their completion status. The viewer program is also used to submit, pause and cancel jobs and set the availability of Distributed OrcaFlex processing clients.
- A **console** program (dofcmd.exe) that performs some of the viewer functions from a console window or batch file. The console program is installed with the Viewer program.

The Distributed OrcaFlex client program runs with a low operating system priority to minimise interference with other programs running on the same machine. The user should not notice a significant impact on responsiveness due to Distributed OrcaFlex. By default, the client program will run as many jobs in parallel as there are available processor cores on the computer. However, this number can be reduced at runtime by using the Distributed OrcaFlex viewer or console program. The default maximum number of cores is set through a registry setting.

2 What's New

Version 6.0d

- A new Orcina logo.
- Adds a small delay between starting DOF Client processes running on the same machine (computers with large core counts may start more than one DOF Client). This is to allow more time for the DOF Server to add each client as it connects.
- Supports a new external function attribute ('*CanResumeSimulation*') that was added in OrcaFlex 10.3a to identify functions that do not save their processing state correctly (often they are using code provided by a third party). This means the simulation cannot resume from a partially run state and consequently DOF will not auto-save the model. If DOF is required to pause or move one of these models then the DOF Client will ignore this and continue running the simulation. This only applies to models run with OrcaFlex DLL version 10.3a or later, with earlier DLL versions the simulation will paused, moved or auto-saved as normal, but it will not resume correctly if using such an external function.

- The default setting for writing the job list to a file has changed from automatically writing this file to not writing this file unless a registry setting ('DisableListPersistence') is set to false. The job list can be large and impose a performance penalty when the DOF Server is busy. It is also a potential source of error when restarting the DOF Server. A consequence of this change is that when the DOF Server restarts, any record of pending jobs is lost and they will need resubmitting.
- The DOF Server also used to write completed job details to a completion log ('JobLog.txt'), this log will not be written automatically unless specifically enabled with the 'DisableCompletionLog' registry setting.

Version 6.0c

- Bug fix: Sometimes, in the event of an error, the DOF Server would produce a cascade of error reports that made the DOF Server unresponsive for a while. This is now resolved, the DOF Server reports all errors to the DOFServer.log without generating any further error files.
- Bug fix: When sending jobs to DOF using dofcmd, a value of 0 (= disable autosaving) for the Autosave interval was not allowed, when this is actually a valid interval.
- Bug fix: If the DOF Server is restarted while some jobs are still running on clients, then those jobs could end up being cancelled rather than re-added to the job list to continue as normal.
- Bug fix: If jobs were submitted whilst the DOF Server was already distributing jobs to DOF Clients then the scheduler ramping feature was re-initiated. Now the ramping feature only starts if the jobs are added when the DOF Server is idle.
- In the client list view of the DOF Viewer, the list columns are now resizeable.

Version 6.0b

- Bug fix: At startup, a DOF Client machine running multiple client processes would appear in the DOF Viewer with a low processor count rather than the true total for the machine. A related problem was that setting the client's processor count to 'None' through the DOF Viewer had no effect.
- Bug fix: If an error occurred in the DOF Server caused by a repeating problem (such as a communication error), then multiple error reports were created that could fill the C:\ProgramData\Orcina\DOF directory, and prevent the DOF Server from responding to the DOF Clients and Viewer.
- Bug fix: When adding small job batches, these may be all scheduled and queued locally to a small number of DOF Clients, leaving other clients idle that should be sharing the processing.

Version 6.0a

- The major change in this release is the ability to have more than one DOF Client program running on the same physical computer. This enables Distributed OrcaFlex to utilise all the processor cores on a computer that has processor groups – generally large capacity servers. A DOF Client process starts per processor group to give full utilization, and optionally the number of DOF Clients can be set higher than this. This will also benefit models using Python external functions or post-calculation actions as there will now be a Python interpreter per DOF Client process, reducing the impact of the processing bottleneck the Python Interpreter introduces.

- Jobs can now be manually paused and resumed from the DOF Viewer. A paused job will remain so until resumed by the user from the DOF Viewer.
- DOF Server functions that automatically move jobs between clients have been removed. This includes forcibly pausing and moving one user's jobs to make way for another user's, and moving jobs from slower to faster computers towards the end of a batch run. In the previous version of DOF these functions were disabled by default. These functions offered only limited benefits and in some cases unnecessarily moved jobs. Removing the functions allows for a more streamlined server. The new manual pause and resume feature can be used to achieve the same ends.
- You can optionally choose to set up the DOF Server to operate as a straightforward batch processor (using a registry setting). In this mode, processors are not shared between users, instead jobs are run in the order they are submitted to DOF.
- Each DOF Client has a small queue for buffering pending jobs sent by the DOF server. This reduces the time between finishing one job and starting another. This will improved throughput for shorter jobs.
- Processing of new job batches is ramped up slowly (over about 2 mins). This smooths the job throughput by preventing a spike of jobs saving and starting at the same time. This feature is enabled by default but can be disabled using a registry setting.

3 Installation

For correct operation, all parts of the Distributed OrcaFlex system should be at the same version. All the Distributed OrcaFlex programs are installed from the single Windows Installer file **DistributedOrcaFlex.msi**. Simply double-click this file to begin the installation. The Server, Client and Viewer (including dofcmd) programs can be separately selected for installation. Some configuration data will be required as part of the install process, described below for each program. In most cases if a setting is missing a default value is provided which will work perfectly well.

3.1 Installing the Server and Client Program on the same computer

The DOF Client can use all the available cores on a machine when processing jobs potentially starving the server program of CPU time. For this reason we do not recommend installing these two programs on the same computer to avoid conflict. If you must do this then you should reduce the default thread count of the DOF Client by 1-2 (or by 2-4 if hyper-threading is enabled) using the DOFMaxNumberOfJobs registry setting of the DOF Client described later.

4 Server Program

The Distributed OrcaFlex server is a Windows service, an application that starts automatically and runs in the background. A short interruption of this service should not affect the overall running of Distributed OrcaFlex, a longer interruption will prevent users submitting jobs and stop queued jobs from starting. Jobs already running should complete, but may be reported as failed by the server on restart.

4.1 Hardware Requirements

The computer should be running Windows 7 or later and must be connected to a network. This machine can also be used to run a Distributed OrcaFlex client if required, but the processor count on this Distributed OrcaFlex Client should be reduced by 1 (physical core, 2 logical cores) to allow spare CPU for the DOF Server, but we do not recommend this configuration if there are many DOF Client machines. A fast processor is desirable.

4.2 Settings

All of the configuration settings for the Distributed OrcaFlex server program are in the system registry under the following key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DOFServerService
```

This key contains the service configuration settings, which should not be changed. It also can contain these user configurable settings:

- *Port*
- *DeleteJobsFromListAfterDays*
- *DeleteClientsFromListAfterInactiveDays*
- *DisableCapacitySharing*
- *DisableRamping*
- JobSchedulingIntervalSeconds
- *WakeOnLANEnabled*
- *DOFLogLevel*
- *DisableListPersistence*
- *DisableCompletionLog*

Incorrect settings may prevent Distributed OrcaFlex from running correctly. If any of these settings are changed then the DOF Server service will need to be restarted to read the new configuration.

Port: REG_DWORD

Port is the TCP/IP port number that Distributed OrcaFlex uses to communicate with the Distributed OrcaFlex client machines. If this setting is missing then the default of 53412 will be used. If this is changed from the default then remember to modify the port setting for all the DOF Client and DOF Viewer machines as well.

DeleteJobsFromListAfterDays: REG_DWORD

Completed jobs remain in the queue after completion so that their status can be viewed for several days. This setting specifies the integer number of days a job is retained. If this setting is missing then the default is 3 days. The maximum number of completed jobs allowed in the jobs list is 10,000 and surplus completed jobs are removed from the list regardless of the value of this setting.

For long term auditing, when a job is deleted from the queue a copy of the job information is made in the file *%COMMON APPDATA%\orcina\dof\JobLog.txt,* where *%COMMON APPDATA%* is the common application data folder (normally C:\ProgramData\). The file is a comma separated text file with one line for each job. Note that filenames containing commas will be enclosed in double quotes.

DeleteClientsFromListAfterInactiveDays: REG_DWORD

The DOF Server keeps a record of recently connected DOF Clients. DOF Clients remain listed in the DOF Viewer even if they do not connect (the client will be marked as 'Unavailable'). This setting is an integer specifying the number of days an 'Unavailable' client will remain in the list before it is removed completely. If this setting is missing, then the default of 4 days will be used. If a client subsequently reconnects, it will reappear in the client list.

DisableCapacitySharing: REG_DWORD

The DOF Server allocates DOF Client resources so that each user with waiting jobs gets an equal share regardless of when they submitted jobs. This is the default operation but if you wish you can disable this feature by adding this setting with a non-zero value. In this case jobs are processed in the order they are submitted.

DisableRamping: REG_DWORD

When a new job batch is submitted the jobs are now gradually distributed to the DOF Clients in increasing numbers at each scheduling action – the interval between scheduling actions also starts long and reduces. This spreads the start times (and therefore the end times) of the jobs and prevents all the jobs trying to save at the same time. You can disable this behaviour (which may be better if you have few clients or processor cores) by adding this setting with a non-zero value.

JobSchedulingIntervalSeconds: REG_DWORD

This setting is an integer number of seconds (1 to 15) between scheduling actions in the server. At each interval a number of jobs (up to the capacity of the client) are assigned to each client with available processors, starting at the top of the client list as seen in the DOF Viewer. If this setting is missing, the default interval is 2 seconds. Increasing this interval will have the effect of throttling the throughput of jobs which may be desirable if this is impacting other programs and services. If ramping is enabled (the default) then the scheduling interval starts at the maximum of 15 seconds and gradually reduces to the value of this setting.

WakeOnLANEnabled: REG_DWORD

If you are not using this functionality and don't want additional WOL packets on your network or you simply don't want machines that are hibernating or in standby being woken up, then you can disable it by setting this to 0. By default Wake on LAN will be enabled. For WakeOnLAN to work, the client machines must have this feature enabled in their BIOS (it may not be by default).

DOFLogLevel: REG_DWORD

By default, the server writes a log of its activity to *Common AppData* (normally *C:\ProgramData*) *\orcina\dof\DOFServer.log* . This file is limited to 10MB and will automatically rotate when this limit is reached. The logging detail can be limited to error information only by setting DOFLogLevel to 0.

DisableListPersistence: REG_DWORD

If this setting is absent then by default, the server will **not** periodically save the list of pending and completed jobs to *Common AppData* (normally *C:\ProgramData*) *\orcina\dof\DOFServer.job* To enable this file being written then add this setting with a value of 0 (false).

DisableCompletionLog: REG_DWORD

If this setting is absent then by default, the DOF Server will **not** log completed jobs to *Common AppData* (normally *C:\ProgramData*) *\orcina\dof\JobList.txt*. If you wish this log to be written then add this setting with a value of 0 (false).

4.3 Job Allocation

Jobs are allocated to clients in the order they are listed in the Viewer. At each scheduling interval the server works down this list and allocates jobs to clients with spare capacity. The number of jobs allocated at each interval is a maximum of double the number of processors for that client (the client has its own buffer queue).

There is a time overhead between a job being scheduled to a client, the client loading and starting that job and reporting back to the server. For short simulations and static analyses the actual job calculation time can be less than this overhead and jobs may not be scheduled fast enough to fully utilise the capacity of the clients. Indeed the job maybe completed before any progress update can be sent to the Viewer.

The server aims to share the available processing capacity equally between users with waiting jobs. When capacity becomes available, pending jobs from the user with the lowest share of the running jobs are allocated first. This feature can be disabled however.

5 Client Program

This program runs as a Windows service running jobs allocated by the server as a background process. It should not adversely affect the performance of other programs running on that computer, although if the user is running other processor-intensive foreground programs then these will take priority over the Distributed OrcaFlex jobs running on that client.

5.1 Hardware Requirements

The computer should be running Windows 7 or later, and must be connected to a network and have read and write access to an area of file storage common to all of the Distributed OrcaFlex client machines. For best performance, we recommend you look at the hardware requirements for OrcaFlex on our website here: (but disregard the graphics requirement).

5.2 Installation

During the installation you will be prompted for a set of credentials for the Client service to run under. We recommend that you create a new specific user, for example '`DOFUser`', which can then be used for all installations of the Distributed OrcaFlex client. This user should be created before you begin the installation and only be used for the Distributed OrcaFlex Client service.

This user must have the *Log on as a Service* privilege and have been granted sufficient rights to be able to read and write to the shared areas of the network filing system that jobs may be submitted from, including the location of the default OrcFxAPI.dll files. The *Log on as a service* privilege is normally set by group policy on the domain controller. Using this method it is possible to maintain segregated team network storage areas to which only the Distributed OrcaFlex user has overall access.

5.3 32/64 bit installation

OrcaFlex 9.6 introduced 64 bit versions of the OrcaFlex executable. In order to use the 64 bit OrcFxAPI.dll, a 64 bit version of the Distributed OrcaFlex client program must be used. The installation program gives you the option of running either the 32 bit or 64 bit version of the client.

If you run the 64 bit version of the client then you will not be able to process jobs using 32 bit versions of OrcFxAPI.dll. So if you need to process jobs with OrcaFlex 9.5 (the minimum supported version) then all your clients must be the 32 bit version. Once you are prepared to commit to OrcaFlex 9.6 and later you can start running the 64 bit version of the client.

If some of your computers are still 32 bit then it is possible to operate a mixed installation where you have some clients running 32 bit and others running 64 bit. In order for a mixed installation to work, the clients must be able to locate a version of the OrcFxAPI.dll that matches the architecture of the host process - a 32 bit client can only load 32 bit DLLs, and a 64 bit client can only load 64 bit DLLs. This issue is dealt with by means of directory naming convention. You need to organise the directory that contains the OrcFxAPI DLLs like this:

```
MyOrcFxAPIDirectory
    Win32
        OrcFxAPI.dll   (the 32 bit version)
    Win64
        OrcFxAPI.dll   (the 64 bit version)
```

So long as the two OrcFxAPI DLLs are arranged in this way, the 32 bit DLL inside a folder named Win32 and the 64 bit DLL inside a folder named Win64, the client program will load the appropriate DLL regardless of which version of the DLL is specified when submitting jobs or through the *DOFDefaultDLLFileName* setting.

5.4 Settings

The settings for the client program are stored in the system registry of the local machine, under the following key:

```
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Orcina
```

(On a 32 bit machine the key is: `HKEY_LOCAL_MACHINE\Software\Orcina`)

This key can contain the following entries, some of which are shared by the Viewer Program:

DOFPort: REG_DWORD

DOFPort is the TCP/IP port number that Distributed OrcaFlex clients use to communicate with the Distributed OrcaFlex server. If this setting is missing then the default of 53412 will be used.

DOFServerAddress: REG_SZ

DOFServerAddress is the TCP/IP network address or hostname of the Distributed OrcaFlex server. This **must be set** correctly for the Distributed OrcaFlex clients to locate the server.

DOFMaxNumberOfJobs: REG_DWORD

DOFMaxNumberOfJobs specifies the maximum number of processors that DOF can use on this machine. If more than one DOF Client is running on the same machine then this number is shared amongst them. If this setting is missing then the total number of processor cores is used (logical cores if hyper-threading is enabled). The processor count can't be increased above this amount in the DOF Viewer.

DOFMaxThreadCountPerProcess: REG_DWORD

If the DOF Client machine architecture has processor groups then, by default, the DOF Client service will start one DOF Client process per group and set the thread affinities accordingly. You can increase the number of DOF Client processes created by assigning a reduced value for this setting. For instance a 40 core server with 4 processor groups will by default get 4 DOF Client processes with 10 threads each. If this registry setting is present with a value of 5, then 8 DOF Client processes will be started with 5 threads each. Beware, creating too many DOF Clients may hamper the performance of the DOF Server as each client independently sends progress messages to the server.

DOFStartClientDisabled: REG_DWORD

Setting DOF*StartClientDisabled* to any value other than 0 will cause the client to start-up disabled i.e. not accepting jobs. If the setting is missing, then the client will start-up enabled. The client can then be enabled in the DOF Viewer when required.

DOFMachinePriority: REG_DWORD

The machine priority is used by the scheduler to help sort the clients when allocating jobs. Normally the fastest machine is sorted to the top of the list but if this is not desired (e.g. the preferred client is a large capacity server that happens to have a slower processor) then this setting can be used to alter the sort order. This setting is an integer between 1 and 3, 1 is the highest priority. A default value of 2 is used if this setting is absent. The priority can also be changed from the DOF Viewer.

DOFSaveLoadLimit: REG_DWORD

The client can limit the number of simultaneous job load and save operations it performs to limit the peak load on the file system. The minimum value is 1, there is no maximum value but a value greater than the processor count of the DOF Client will have no practical effect. By default this value is set to unlimited. Processing threads that are waiting to save are not available to run new jobs. The maximum limit on the number of jobs that can be loaded or saved concurrently is determined by the capacity of the network and filesystem.

DOFLogLevel: REG_DWORD

By default, the client writes a log of its activity to *Common AppData* (normally *C:\ProgramData*)*\orcina\dof\DOFClient.log*. This file is limited to 10MB and will automatically rotate when this limit is reached. The logging detail can be limited to error information only by setting DOFLogLevel to 0. When multiple DOF Clients are running on the same computer then each client saves its own log, with the machine name and process ID as part of the filename.

6 Viewer program

This is the main interface to Distributed OrcaFlex and allows users to submit jobs, pause and cancel jobs, view job progress, and control some client settings. When open, the Viewer receives status updates from the DOF Server on the current list of pending, running, and completed jobs and active clients. For long job lists, this can consume a significant amount of the DOF Server processing time, so avoid keeping many Viewers open in this situation. When minimised to the system tray the Viewer disconnects from the DOF Server.

6.1 Hardware Requirements

The computer should be running Windows 7 or later, and must be connected to a network and have read and write access to an area of file storage common to all of the Distributed OrcaFlex client machines.

6.2 Settings

The settings for the viewer program are stored in the system registry of the local machine, under the following key:

```
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Orcina
```

(On a 32 bit machine the key is: `HKEY_LOCAL_MACHINE\Software\Orcina)`

Changes to these settings require the service to be restarted to read the new configuration. This key can contain the following entries, some of which are shared by the Client Program:

DOFPort: REG_DWORD

See description for the Client program's DOF*Port* setting.

DOFServerAddress: REG_SZ

See description for Client program's DOF*ServerAddress* setting.

DOFAutoSaveIntervalMins: REG_DWORD

DOFAutoSaveIntervalMins defines default number of minutes between backup saves. The auto backup save allows any other client to pick up the job if one of the clients fails. The default value is 60 minutes. This can be changed in the DOF Viewer when submitting jobs.

DOFDefaultDLLFileName: REG_SZ

DOFDefaultDLLFileName is the full path to the OrcFxAPI.dll file that should be used if a DLL file is not specified when submitting jobs. The DLL must be on a network drive that all of the clients can read. For systems with a mix of 32 bit and 64 bit clients, the two versions of the DLL should present in a directory structure described in section 5.3, and the full path to one of these should be the value of this setting. The value of this setting is displayed in the Settings tab when submitting jobs.

6.3 Starting

The viewer program is started by running the executable DistributedOrcaFlex.exe, from a shortcut on the desktop or Start menu. The default installation will ensure that this is started automatically when a user logs in.

6.4 Using the Viewer Program

OrcaFlex Clients Page

This page shows all the client machines that the server is aware of, in order of highest priority first.

Machine, Host Name, IP Address: Client machine identity. Double-clicking on a client node will expand the node to show each client process on the same machine (double click again to hide the detail). The name of each client comprises the machine name and the process id of that client.

Version: The version of the Distributed OrcaFlex client software running on that machine.

Processors: Total number of processing cores on that machine available to the DOF Client(s).

Available Processors: The current number of processors available to the DOF Client to run OrcaFlex jobs.

Jobs Running: The number of OrcaFlex jobs currently running on that machine.

Status: The availability of that client. See the Standby and Hibernation section below.

Machine Priority: This is the ranking of that client used by the server program. Clients are ranked first by priority (which can be set in the Viewer) – lowest priority value first, and then by the machine speed index (calculated by the client itself) – highest value first.

The *Available Processors* and *Machine Priority* values for any active client can be set by highlighting the client in the list and then modifying these in the pop-up menu. Setting *Available Processors* to ‘None’ disables that client. If the number of available processors is set to less than the number of jobs running then the surplus jobs are moved back to the waiting queue for another client to run (most recently started job first).

The machine priority value can be used to alter the ranking of clients if it is not desired to have the fastest machine at the top of the list (for instance to push a slower, but larger capacity server to the top).

Jobs Page

This page lists all the jobs, both completed and pending. The list can be filtered by user and job status. Job progress and status are updated by the DOF Server at regular intervals, however if the system is busy then these updates may be delayed or skipped to prioritise communication with the processing clients.

File: Each Distributed OrcaFlex job is identified by its file name.

Analysis: the calculation to be performed, Statics only or Dynamics.

DLL Version: The version of the OrcFxAPI that is being used to run the job. It is important that this version is the same as or higher than the version of OrcaFlex that was used to create the dat file. Results may be affected if an older version of the OrcFxAPI is used as it may not support all of the features required by the dat file. The version is only displayed once the file is loaded by a client.

Owner: This column shows the name of the user who submitted the job to the queue.

Status: Each job has a status, which will be one of Pending, Starting, Running, Post Calculation Actions, Completed, Cancelled, Paused or Moving. Jobs that were cancelled by a user will not be restarted. Jobs that failed due to a licencing problem (e.g. not enough licences available) will be restarted on the next available machine.

Completion Time: This shows either the actual time the job completed or, if it is in brackets, the estimated time to completion. It should be stressed that this is only an estimate and will change if machine loading changes or the job is paused or moved to another machine.

Last Run On: The machine the job is or was most recently running on. Where there is more than one DOF Client processes running on the same machine then the machine name also includes the process id on that machine.

Auto Save (mins): Displays the interval in minutes between successive auto-saves.

Status bar information

The information in the status bar at the bottom of the DOF Viewer window shows the connection status (and version) of the DOF Server, the total currently available processor count and the potential maximum processor count for all the DOF Clients. Running and Waiting job counts are also displayed. When the first running job reaches completion an additional 'Completion rate' item will appear which gives an estimated the rate of job completion.

Submitting Jobs

OrcaFlex jobs are submitted to Distributed OrcaFlex by clicking the *Add Jobs* button on the toolbar or pressing the *Insert* key, which pops up a *Submit* dialog. From the *Jobs* tab, click *Add* and select one or more files from a networked drive (OrcaFlex dat or sim files must be on a network drive so that all clients will be able to locate them) then choose *Static* or *Dynamic* analysis from the option at the bottom of the dialog, and finally click *Open*.

On the *Settings* tab, the following options are available.

- **Auto save interval (mins)**: Distributed OrcaFlex will attempt to save the current simulation state at this time interval so that if a machine fails another machine can pick up the job from the last simulation saved. A value of 0 will result in no backup saves.
- **OrcFxAPI filename**: In order to run a simulation the user must specify the location of the OrcFxAPI.dll that the clients must use – again this must be on a network drive in a location to which all the client machines have access. If this box is left blank then the default OrcFxAPI.dll will be used, this default was specified during installation as a registry setting.

When you have selected all the files, click on the *Submit* button.

If a partially completed simulation file is submitted to the jobs list, Distributed OrcaFlex will complete the simulation.

Jobs can also be submitted by simply dragging them onto the Jobs List from Windows Explorer, which will bring up the *Submit* files dialog.

Resubmitting Jobs

Jobs can be resubmitted to the job queue by selecting the job or jobs in the jobs list view and clicking the **Resubmit** button or pressing *CTRL-R*. The jobs are resubmitted with the same parameters as when they were first submitted.

Cancelling Jobs

To cancel a job, select the job or jobs and click the *Cancel Jobs* button or press the *Delete* key.

Pausing and Resuming Jobs

Pending or running jobs can be paused from the DOF Viewer, select the job(s) then click the pause icon, or right-click to show the pop-up menu and select 'Pause selected jobs'. These jobs will remain paused until you resume them by selecting the same job(s) and clicking the 'Play' icon or choosing 'Resume selected jobs' from the pop-up menu.

Viewing Jobs with OrcaFlex

With a job selected, you can view the data file or the simulation file by clicking the relevant button on the toolbar. If AutoSave is enabled then the '.sim' file may be the partially completed simulation. For this to work '.dat' and '.sim' files should have a file association with OrcaFlex, and must exist for the selected job. An alternative to using the toolbar buttons is to press *ENTER* or *CTRL-ENTER* respectively.

***Note**: These functions are also available by clicking the right mouse button on a selected job to display the pop-menu.*

Clear Completed Jobs

A user who has completed jobs in the job list can clear these from the Viewer (and the Server job list) by clicking this button. This button is only enabled for the current user with jobs at this status, jobs for other users can only be deleted from the DOF Viewer running under their user account.

6.5 Enabling and Disabling

If the DOF Viewer is running on the same machine as a client program then that client can be enabled and disabled by right clicking the OrcaFlex icon in the task bar and selecting the *Available Processors* menu item. From this menu you can disable the client by selecting *None* or enable it by specifying how many processor cores you wish to make available to Distributed OrcaFlex. For other clients, this can also be done through the DOF Viewer itself.

If the number of processors cores available is reduced or the client is disabled, then after a short period of time, jobs running on the local machine will be moved back to the waiting queue.

6.6 Client program as an Application

In some circumstances, running the Distributed OrcaFlex client as a Windows service is unsuitable, so the ability to run as a normal application has been maintained. Although in this configuration a user must always be logged in for the client to be active.

If you wish to do this, you must first stop and disable the OrcaFlex Client service that the installer configured and start the client each time a user logs in by using the following command:

```
DOFClient.exe -RunAsApp
```

The client will then use the currently logged on users credentials to access the network.

6.7 OrcaFlex Licences

A Distributed OrcaFlex client will only ever claim one OrcaFlex licence regardless of the number of jobs running on that machine. In addition to this, a user physically sitting at that machine will be able to share the same single licence if running OrcaFlex. If no jobs are running, the client will not claim a licence.

However, if you log into the machine via Windows Remote Desktop, then a second licence will be required for that user to run OrcaFlex.

An alternative is to use a product like VNC (www.realvnc.com) which results in the same licence usage as if you were sitting at the machine, i.e. only one OrcaFlex licence is required for both the user and the client to operate (they are in the same user session).

7 Console Program

The console program can be used for submitting jobs for processing, listing the status of queued jobs, or setting the priority and availability of a named Distributed OrcaFlex client from a command line or script.

7.1 Hardware Requirements

The hardware requirements for `dofcmd.exe` are the same as that for the viewer program.

7.2 Settings

The settings for `dofcmd.exe` are stored in the system registry. The `dofcmd.exe` program shares its settings with the viewer program (see 6.2 Settings). For ease of use, you may wish to either add the Distributed OrcaFlex installation folder to your PATH environment variable or copy `dofcmd.exe` from there to a location already on your PATH.

7.3 Using the Console Program

Just typing the following at the console with no parameters will display help:

```
dofcmd
```

To list jobs in the queue, type the following command into a console window:

```
dofcmd -list
```

The output from this command is the jobs queue in text format. There is one job per line, with each line consisting of comma separated values containing the following fields: Job ID, Simulation file name including full path, Owner, Status, Start Time, Completed Time, Name of machine last run on, IP address of machine last run on, OrcFxAPI DLL version and Status string. Note that filenames and any status text containing commas will be enclosed in double quotes.

The job ID is a unique identifier used by Distributed OrcaFlex; the other items are explained in more detail in section 6.4 (Using the Viewer Program).

To add jobs for processing use the following command:

```
dofcmd -add [OPTIONS] <filename1> … <filenameN>
```

will add the files listed to the Distributed OrcaFlex queue using the following [OPTIONS]:

`-dynamics`	Run a dynamic analysis on the jobs added (this is the default).

`-statics` Run a static analysis on the jobs added.

`-dllname=<filename>` Where <filename> specifies the full path of the OrcFxAPI DLL. If this option is missing then the default DLL will be used.

`-autosave=<n>` <n> is the autosave interval in minutes. If this option is missing then the default autosave time will be used.

`-wait` Wait until all jobs have completed before returning.

The files to be added can be OrcaFlex data or simulation files (*.dat, *.yml or *.sim). Alternatively, a text file containing a list of OrcaFlex data or simulation files (with the full path) can be specified.

A return value of 0 means the call was successful, a value of 1 means that an error message was generated and sent to *stdout.*

Jobs can be removed (cancelled) with the following command:

```
dofcmd -cancel <filename1> … <filenameN>
```

The files to be removed are specified in the same way as for adding files. The program will match the specified files with jobs in the joblist that are running or pending, and cancel these.

To change the number of available processors for a Distributed OrcaFlex client, use the following commad:

```
dofcmd -machine=<name> -processors=<n>
```

Where:

`-machine=<name>` <name> is the name of the client machine as it appears in the Viewer program 'Machine' column of the OrcaFlex Clients page.

`-processors=<n>` <n> is the number of processors available for running jobs. Use 0 to disable the client (the equivalent of choosing 'None' in the Viewer program or from the task bar icon).

To modify the *Machine Priority* value of a client, to change its relative position in the scheduling queue, use this command:

```
dofcmd -machine=<name> -priority=<n>
```

Where:

`-machine=<name>` <name> is the name of the client machine as it appears in the Viewer program 'Machine' column of the OrcaFlex Clients page.

`-priority=<n>` <n> is the priority value to use when the scheduler is ranking clients relative to each other. The highest priority is 1, the lowest 5.

Completed jobs can be cleared from the server job list with the following command:

```
        dofcmd -clearjobs=<owner>
```

Where `<owner>` is the name of the job owner (as displayed in the DOF Viewer under the 'owner' column). Alternatively, use `all` to clear every completed job from the job list.

8 Standby and Hibernation

In order to reduce power consumption, The Distributed OrcaFlex server is capable of waking machines running the Distributed OrcaFlex client program from Windows Standby or Hibernation states when they are required to run jobs. In order for this to function correctly, the client machine's BIOS needs to be configured to allow *Wake On Lan* (WOL) and the network card's *Power Management* needs to be configured to allow it to bring the machine out of standby state (Network Card, Properties, Power Management).

The OrcaFlex Clients page in the viewer reports the following states:

Active:	Client is connected to the server and is ready to accept jobs.
Sleeping:	Client is hibernating (or possibly switched off).
Waking:	Server is attempting to wake the machine from hibernate state.
Unavailable:	Client has been shutdown, or the server was unable to wake it.

If WOL is disabled by using the Distributed OrcaFlex Server setting, then clients will never go into the Sleeping or Waking states and will be either Active or Unavailable.

Note that the server cannot be allowed to hibernate or go into standby, and must be running at all times.

9 Points to Note

9.1 External Functions and Post Calculation Actions

If the OrcaFlex models being processed by Distributed OrcaFlex use either Python External Functions or Post Calculation Actions then each Distributed OrcaFlex client machine needs the required version of Python installed. Similarly for Post Calculation Actions using command scripts then programs used by the script need to be present on each client machine. Note, MATLAB can't be started by a Distributed OrcaFlex client service since it tries to open a console on the desktop – and this is not possible for an application running as a service.

If the model file has the 'Skip save' setting enabled then the simulation file for that model will not be saved after the post calculation action has completed, also any auto save setting will be ignored.

When running on computers with a large processor count then Python processing will be slow as the Python interpreter is effectively single threaded. There is one Python interpreter per DOF Client process, so it will be beneficial in this case to configure the client machine to start more DOF Clients. See the 'DOFMaxThreadCountPerProcess' registry setting for the DOF Client above.

9.2 OrcaFlex Upgrades

Installing an upgrade to OrcaFlex on machines that also have Distributed OrcaFlex installed does not update the default OrcaFxAPI.dll (used when processing jobs). The location of the default OrcFxAPI.dll is defined on each of the client machines by the registry setting *DOFDefaultDLLFileName* which is a string containing the full UNC path to the DLL.

To update the default DLL used by Distributed OrcaFlex, you need to manually copy the new version of the DLL to the location specified by the *DOFDefaultDLLFileName* setting. Naturally, if your setup includes both 32 bit and 64 bit clients then you need to copy both 32 bit and 64 bit DLLs.

If the location of the default OrcFxAPI.dll needs to be changed, then it is very important that all the client machines running the DOF Viewer have their registry setting updated to reflect the new location. This setting can be seen in the DOF Viewer in the 'settings' tab when submitting jobs.