



## OperaFLO White Paper - Application Programming Interface

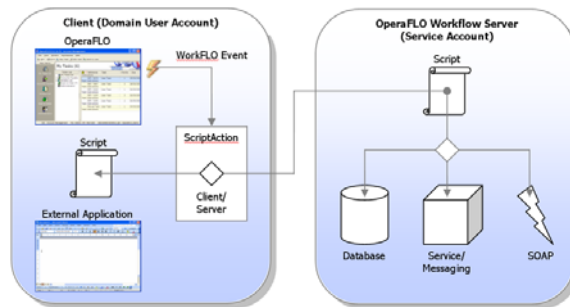


### Interoperability & Integration

OperaFLO provides a comprehensive set of capabilities for integration and interoperability with external systems. A range of technologies are available to meet the needs of the integrator.

### Extending OperaFLO

OperaFLO uses an event-driven workflow model. Each defined process exposes a key set of events that can be configured to trigger actions. OperaFLO provides a comprehensive set of actions that can be used without programming or scripting to model complex processes within the system. These actions can change the state of the process on which they are executed, create tasks for users, set deadlines etc. The out-of-the-box suite of actions all execute within the context of the OperaFLO system.



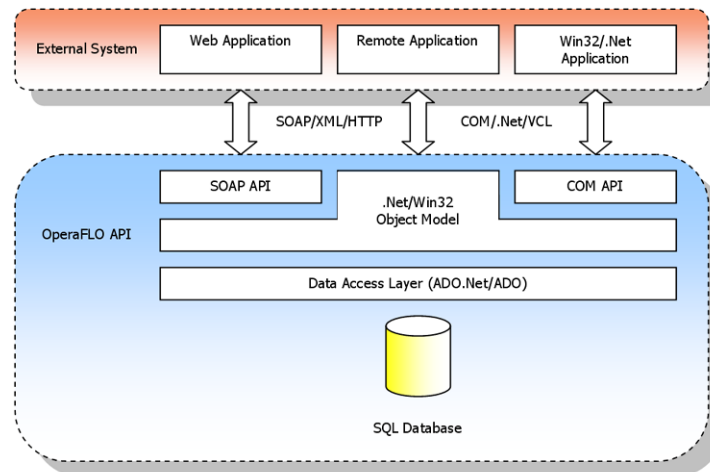
To reach outside of the OperaFLO system there is an Action that can execute scripts developed in any ActiveScript compliant language (eg VBScript, JScript, DelphiScript etc). Scripts attached to these Actions can be configured to execute either synchronously on the client machine or asynchronously on the OperaFLO workflow server.

Client-side scripts can be used to integrate OperaFLO with any line-of-business application in use on client machines such as Microsoft Office. In fact, any application or library that supports either an automation or command-line interface can be controlled via OperaFLO client-side scripts.

For integration with back-end systems, server-side scripts are executed by the OperaFLO workflow server allowing the lock-down of the security context within which the scripts execute. This provides a secure environment for OperaFLO to integrate with back-end databases, services or call into remote systems via SOAP or DCOM. Server-side scripts are executed asynchronously through queue-based processing. This allows the OperaFLO workflow server to monitor overall processing load and manage script execution accordingly.

## The OperaFLO API

Where more complex integration solutions are required, OperaFLO provides a layered API (Application Programming Interface) that exposes key functionality for consumption across a broad range of technologies.



By layering the API, OperaFLO provides integration on-ramps at various levels which range in functionality and complexity. The SOAP interface for example, provides a subset of the overall API functionality, concentrating on key end-user functionality and providing a high-level abstraction of the underlying object model. For more complex projects the underlying API can be accessed as a .Net Assembly, exposing a broader range of OperaFLO functionality at a finer level of detail and complexity.

## SOAP & COM APIs

The SOAP & COM interfaces allow platform-neutral, standards-compliant integration with OperaFLO. The SOAP interface allows access to key system functionality via HTTP and XML and the COM interface provides equivalent functionality for in-process or LAN based integration clients.

Both interfaces expose a subset of the overall OperaFLO API focusing on the key data and functions of the system and hiding more complex details:

### *Querying System Metadata*

Since the structure of data & workflow processing within OperaFLO is customer defined, access to the system metadata is fundamental to working with data in the system. Using the simple APIs, detailed information relating to the data schema and process models can be obtained for use by the integration client.

## *Querying, Creating & Updating Instance Data*

Read/write access to all processes within the OperaFLO system provides for comprehensive data integration with external systems. Processes can be queried, created and have their state changed. Events are fired by the API in the same way that they are by normal use of the OperaFLO client application, so configured workflow processes can be initiated by external systems with ease.

## *Task Management*

User tasks can be generated by actions configured on OperaFLO events but the SOAP interface also provides access to tasks and task lists. Tasks can be created for specific users or groups and any task list can be queried honouring any permissions applicable, given the context in which the API is accessing the system.

## *Access To Documents*

The SOAP interface exposes the basic document management capabilities associated with a process. Documents can be attached to a process, queried and downloaded by a remote system. File conversions can be applied where applicable, for example many native document types can be requested as PDF files to broaden the audience for their content.

## **Object Model**

Underpinning the SOAP interface is the OperaFLO Object Model. This is a comprehensive library that exposes OperaFLO functionality at a lower level as a .Net Assembly. The library has been architected to allow a great deal of customisation and for usage in multiple types of hosts such as web-applications, services or client-side applications.

To find out more about FLOvate and any of our products and services (OperaFLO, LicenceFLO or VoiceFLO) please visit [www.flovate.com](http://www.flovate.com)

We will be happy to demonstrate any of our products to you. For either a face to face or a web based demonstration call us on **01473 432550** or email [info@flovate.com](mailto:info@flovate.com)