



# **Interfacing with Avid iNEWS**

Including iNEWS Web Services Version 1.0



## Table of Contents

Overview .....	1
Exchanging Data with iNEWS .....	2
iNEWS FTP Server .....	2
RXNET/TXNET .....	2
Support for MOS Protocol .....	2
iNEWS MOS Gateway .....	4
iNEWS ActiveX Support .....	4
iNEWS Web Services .....	5
Folder Listing .....	5
Queue Listing .....	5
Story Creation, Editing and Deletion .....	5
Messaging .....	6
Searching .....	6
Monitoring Queues for Changes .....	7
Supported Standards .....	8
Web Services Deployment .....	8
Server Requirements .....	8
Client Licensing .....	8
How to obtain iNEWS Web Services .....	8

## **Overview**

The Avid iNEWS newsroom computer system (NRCS) provides journalists with all the tools necessary to plan, create, publish and archive a news broadcast. Scalable from ten to thousands of users working across multiple sites the Avid iNEWS NRCS brings together the newsroom system, editing systems and playout applications in a unified production environment. iNEWS incorporates comprehensive Unicode capabilities, as well as support for iNEWS Command automation-assist including third party servers and integrated newsroom graphics.

iNEWS includes a rich set of features to enable customers and integrators to exchange content with the newsroom system, these features are described in this document. Please note however that whilst this document describes support for details of media assets that might be associated with a story it does not deal with playback control functionality that might be associated with the media devices themselves.

Note that in the descriptions which follow it is assumed that the reader is familiar with the basic iNEWS database structure of folders, queues and stories.

## **Exchanging Data with iNEWS**

iNEWS allows you to exchange data using an industry standard FTP Interface. Data is exchanged in a format known as NSML (News Story Mark-up Language). NSML is an XML-based mark-up language described in "NSML - A Mark-up Language for News Stories", by iNEWS. Using an FTP interface you can perform several story editing and management functions including creating new stories and monitoring the change status of existing stories.

No specific API is required as all communication is via a standard FTP interface, which can be used in two different ways. Note that a much richer set of data exchange features is available using iNEWS Web Services.

### **iNEWS FTP Server**

The iNEWS FTP Server provides conventional FTP access to the iNEWS database. Clients must first login to the server after which standard FTP commands are used to navigate the queues and extract the data requested.

### **RXNET/TXNET**

The RXNET and TXNET implementations differ from the FTP Server in that they are typically associated with specific queues within the iNEWS database. An RXNET implementation for example can be used to listen out for stories that may be posted from a third party application, or another iNEWS system, and then carry out an appropriate action, such as posting them to a specific queue. Similarly a TXNET implementation can be used to send stories automatically to a third party application, or another iNEWS system.

For more information about the FTP interface or NSML please visit Avid Online Support.

## **Support for MOS Protocol**

MOS Protocol is widely used to enable communication between Newsroom Systems and Controlled Devices such as media servers. Typically this is a two-way communication with the Newsroom maintaining an up to date running order on the servers controlled whilst at the same time the servers provide information back to the Newsroom about the state of their media items. iNEWS provides MOS support through a dedicated MOS

Gateway and also supports the use of ActiveX Controls, which are typically used to provide close integration with controlled devices.

### **iNEWS MOS Gateway**

The iNEWS MOS Gateway is a dedicated server-based application which provides an interface between the iNEWS environment and a controlled third party device. It can also be used to provide interfacing to station automation systems. No specific API is required to communicate with the MOS Gateway.

For more information about MOS Protocol please visit the MOS Protocol website.

### **iNEWS ActiveX Support**

ActiveX Controls are typically used to allow direct access to third party devices from within the newsroom client. A good example of this is Avid DekoSelect. By embedding the DekoSelect ActiveX control within the iNEWS client users gain direct control over the content, look and feel of their graphics objects. Once a graphic has been composed to their satisfaction all that is needed is to drag its thumbnail into the iNEWS story. It will be saved as a MOS object and sent to the iNEWS Command system for playout when the rundown is made active.

Other ActiveX implementations include those for third party graphics and subtitling support.

If you are interested in developing an iNEWS ActiveX Integration please contact Avid Professional Services for details of the ActiveX specification.

## **iNEWS Web Services**

iNEWS Web Services provide third parties with a rich set of functions to interface and exchange data with an iNEWS newsroom system. Key functionality includes:

- Folder navigation and listing
- Queue navigation and listing
- Story creation, editing and deletion
- Messaging
- Searching
- Monitoring queues for changes

### **Folder Listing**

The contents of any folder, including the database root node, may be listed. The listing is not itself recursive however third party solutions can easily implement such functionality.

### **Queue Listing**

The contents of any queue can be listed and the contents of any or all of its stories retrieved. Typically a queue listing operation will involve two distinct steps, one to retrieve the form data associated with the queue as a whole then the data for the individual stories themselves. If only a simple listing of story metadata is required then it is possible to retrieve this without needing to read the entire body text for each story.

Note that story data is returned as NSML (News Story Mark-up Language), brief details of which are included in appendix 1.

### **Story Creation, Editing and Deletion**

A story may be "locked" thus enabling edits to be made without any risk that a subsequent operator action would compromise any changes made. A story that has been locked by the web services interface appears to other operators in precisely the same way as any other edit locked story. The story will be shown as locked by the user name which was used to perform the web services login. Once any edits are complete and the story saved back to the server the edit lock is automatically released.

Note that if a story has already been locked by a user or another web services operation then any subsequent request to lock it will fail.

A new story can be created and saved to the server in a single call. An option for the create story function is to retain edit lock on the story, which can then be edited further and subsequently saved. This allows a new story either to be prepared completely in advance, and then written to the server in a single operation, or a potentially "empty" story can be created on the server and its contents added in a subsequent operation.

Stories can be deleted using web services. A separate operation is required for each story to be deleted. Note that a delete operation will fail if the story is locked.

Note that the user credentials employed for web services access must have appropriate access rights to the iNEWS database for these operations to succeed, i.e. the user must be a member of the "Write" group to create and modify stories, and the "Editorial" group to delete them.

## **Messaging**

iNEWS web services provides an interface to the iNEWS messaging system. New messages can be dispatched to any iNEWS user and it is also possible to check for and retrieve any messages that have been addressed to the user associated with the web services login.

## **Searching**

A number of search options are available through iNEWS web services and typically mirror those available through the standard iNEWS client. Searches may be made in both indexed and non-indexed queues and can be restricted to searching the field section of stories, the body section or both. It is also possible to specify a limit as to the number of results to be returned. The use of the Boolean operators "AND" and "OR" is supported for the specification of search criteria.

Search results are not returned as part of the call but are retrieved subsequently in a separate call, which may, of course, return no results in the event that no matches are found. It is also possible to have the availability of any search results communicated via the notification mechanism described in the next section.

## **Monitoring Queues for Changes**

Using iNEWS web services it is possible to monitor selected queues for changes. Queues to be monitored are first registered for change monitoring. Once this is done information concerning any queues that have changed can be reported in one of two ways.

The simpler of these involves the client making a call to query which, if any, of the queues has changed. In this way the database can be polled at regular intervals without needing to re-list the relevant queues. The second method involves the client implementing a notification service which is then registered with the web services interface. Any queue updates result in a notification to the client, indicating that the contents of the queue have changed. Irrespective of the mechanism employed whenever a change is detected the queue listing functionality described earlier can be used to retrieve details of the change(s).

Queues can also be unregistered at any time.

## **Supported Standards**

iNEWS Web Services are based on SOAP 1.1 and are compatible with the most commonly used web services frameworks. Avid has tested the services for interoperability with:

- .Net 2.0
- Sun Metro for Java (JAX-WS)
- Python

Note that since web services are well adopted, iNEWS WS is likely to work with other frameworks and languages like Java Axis 2.0, Python, Perl and others, although these have not been specifically tested by Avid.

## **Web Services Deployment**

Interplay web services are currently deployed as SOAP services running on an Apache server. The web services installer installs the server together with all relevant supporting files. Note that the location of the web services server should be chosen carefully in order to avoid any adverse effect on system operation. Avid Professional Services will be happy to advise as to the most appropriate location for a particular installation.

## **Server Requirements**

Whilst there are no specific hardware requirements to support iNEWS Web Services a modern Windows machine (~2GHz, 1-2 GB RAM) is required, typically configured as a Windows Server. Note that an appropriately higher performance server should be supplied if it is also required to host additional interfaces such as Interplay Web Services.

## **Client Licensing**

No additional licensing is required to add a web services implementation to an iNEWS environment. Each client session however does require an appropriate "api-session" license to be available in the iNEWS system. Customers can order these session licenses from their Avid Sales Representative in the usual way.

## **How to obtain iNEWS Web Services**

The iNEWS web services API is available to developers on request.  
Requests should be made using the web form available at

[http://avid.formbin.com/forms/avid-avid\\_inews\\_web\\_services\\_signup](http://avid.formbin.com/forms/avid-avid_inews_web_services_signup)