



**HAL**  
open science

## A Universal Recorder for the IRCAM Workstation

Bennett K. Smith

► **To cite this version:**

Bennett K. Smith. A Universal Recorder for the IRCAM Workstation. ICMC: International Computer Music Conference, Sep 1990, Glasgow, United Kingdom. pp.1-1. hal-01161131

**HAL Id: hal-01161131**

**<https://hal.science/hal-01161131>**

Submitted on 8 Jun 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Universal Recorder for the Ircam Workstation

Bennett Smith

ICMC 90, Glasgow (Ecosse), 1990

Copyright © Ircam - Centre Georges-Pompidou 1990

---

## Abstract:

The Universal Recorder is a multi-channel recording/playback tool integrated into the environment of the Ircam Musical Workstation. It permits the synchronized recording of parallel data streams of diverse types, and coordinates the simultaneous viewing and editing of these multiple streams. The actual editing is done by different specialized programs, depending on the data type of a given stream. These programs exchange editing messages via the recorder. The recorder can be run from its control panel, or by messages sent from other applications. The recorder is intended as a debugging and development tool, as well as "computerized tape recorder".

## Introduction

The Ircam Musical Workstation (or IMW) is an environment for the generation and manipulation of sound, using two (or more) i860 processors for signal and event processing, and a NeXT computer for user interface. The software environment includes an extensible signal editor [1], and a graphics-oriented environment, ANIMAL [2], for creating and manipulating event lists, DSP processes, interactive control structures, etc. At a certain level, the Ircam Musical Workstation can be considered as a collection of objects, connected by unidirectional data streams. The Universal Recorder is a tool for the synchronized recording and reproduction of the data in these streams. Although a given data stream transmits only one type of datum, different streams can have different data types. One design goal for the recorder was to be able to manipulate these parallel but diverse streams simultaneously. This goal imposes constraints on other parts of the workstation.

The paradigm for the universal recorder is an analog multitrack tape recorder. The control panel has global controls, such as play, record, remote control select, etc., and controls and indicators for each track, such as track name, multitrack edit enable, monitor/playback switch, and record enable.

## The Probe

Each track is associated with a named object called a probe, with a single input and output, analogous to the input and output of one channel of a tape recorder. There are two kinds of probes, one for signals, and one for messages. In either case, a probe can be inserted into an IMW application wherever there is a connection between two objects: the connection is broken, the sending object is connected to the input of the probe, and the output of the probe is connected on the receiving object. As in a tape recorder, each track can be in either record, playback or monitor mode: in record mode, the data read the input are sent to the output, and are also saved, either in memory or in a file. In playback mode, previously saved data are sent to the output, and the input is ignored. In monitor mode, the input is copied to the output, and may also be written into a circular buffer, which serves as a delay line, permitting retroactive recording: when the "start-recording" signal is received, the output of the delay line is saved, effectively moving the start point of the recording back in time.

A probe switches between modes in response to messages, which can come from the recorder or from

other modules in the application, such as an "event detector" to trigger the beginning of recording in response to certain special conditions. There is no fundamental requirement that all probes change mode simultaneously, other than a desire to avoid a confusing mess. For its part, the recorder will only send identical, synchronized messages to all enabled probes. Moreover, the usual way to trigger recording from an event detector is not to send messages directly to the probes, but rather to the recorder's "remote control" probe. This is a probe which is used simply as a place to send messages such as "start-recording", "stoprecording", "pause", making it possible to include a multi-track recording as a sub-module in any IMW application.

## Cooperating applications

An important idea behind the recorder is that it is a separate application; the user places probes in an IMW application, but the recorder communicates with the probes and helps handle the data. The philosophy is that the same tools for data acquisition and inspection should be available to any kind of IMW application (monolithic C program, lisp interpreter, interactive graphic program such as ANIMAL, etc.) To achieve this the functions of a probe must be implemented in each of these environments.

To keep track of probes, the recorder needs to know not only the name of each one, but also enough information to find the application (and under some circumstances the window) where the probe has been placed. This is especially true after reopening a previously saved recorder window: if the user actuates the "show probe" control for a given track, the recorder will have to establish communication with the appropriate application, starting it up if it is not already running, and then get it to display the window containing the probe, which might involve that application also opening a previously saved file. The same thing must happen in reverse if the "show recorder" command is actuated for a remote control probe in another IMV application's window. Fortunately, the NeXT Workspace Manager provides fairly good support for these kinds of operations.

If the user actuates the "edit data" control on a track in the control panel, an editing window for that track opens in the NeXT workspace. The editing window does not belong to the recorder application itself; it belongs to a specialized editing application, which cooperates with the recorder by responding to certain messages. From an object-oriented point of view, one can speak of a track's "edit method". In the case of sound data, this application will be the Signal Editor program, and for messages, ANIMAL, or a more specialized message-editing program such as a MIDI sequencer program. Multiple tracks are viewed and edited using a mechanism of message passing between the recorder and the editing programs. If a number of tracks are enabled for parallel editing in the recorder control panel, actions taken in one editing window will be mirrored in the editing windows of the other enabled tracks. If, for example, the user selects a certain segment of sound in a signal editor window, the signal editor sends a "select:begin-time:end-time" message to the recorder, which sends it on to the other enabled windows.

Multiple-track scrolling occurs when the user moves the cursor: "visible cursor" movement messages are dispatched to the other windows. If the cursor would move out of a window as a result of such a message, the window scrolls once to keep the cursor visible. User actions which change the window itself, such as scrolling and resizing, have no effect on the other windows; these only react when the cursor moves. An editor designed to work with the recorder must therefore support the idea of a cursor, and edit windows should have cursor movement buttons in addition to any scrolling controls. Such an arrangement is less costly than continuous scrolling of multiple windows, which requires much computer power for a tolerable response.

## Some Uses

In addition to the usual recording/playback of musical performances, the recorder is intended to be a tool for development and debugging. The ability to scroll in parallel through multiple streams of recorded data facilitates post-mortem analysis. The possibility of synchronized recording of all inputs of a complex

algorithm, along with the capability for retroactive recording, permits the capture (and perhaps even the reproduction) of elusive intermittent bugs; since the recording can be triggered by a message from another program, elaborate triggering mechanisms can, if necessary, be built up elsewhere, for example in ANIMAL. The inputs to a complex performance-related algorithm, such as a score follower which combines analysis of sound and MIDI inputs, can be saved and reused, permitting development and debugging of the algorithm after the performer has gone home.

---

## References

- [1] G. Eckel, "The SignalEditor of the IRCAM Musical Workstation", ICMC 1990 Proceedings
- [2] E. Lindemann, "ANIMAL: A Rapid Prototyping Environment for Computer Music Systems", ICMC 1990 Proceedings