

# Linux Simulator<sup>TM</sup> v1.0

## Technical Brief

December 2003

# Contents

<b>Introduction to Linux .....</b>	<b>3</b>
<b>What is Linux Simulator? .....</b>	<b>3</b>
<b>Benefits of Linux Simulator’s Advanced Technology.....</b>	<b>4</b>
True Simplicity .....	4
At Last! No More Programming .....	4
Increased Productivity and Cost-Effectiveness .....	4
<b>Powerful Features of Linux Simulator.....</b>	<b>5</b>
Create Linux Simulation Files.....	5
Powerful Compression .....	5
Navigation Bar .....	6
Frames.....	6
Edit Objects .....	6
<i>Mouse Plot</i> .....	7
<i>Instruction Box and Text Field</i> .....	7
<i>Bitmap</i> .....	7
<i>Validation Field</i> .....	7
<i>Hyperlink</i> .....	8
Draw Objects .....	8
Color Palette .....	8
Deploying Simulations as Flash (SWF) Files .....	9
<b>The Superior Results of Linux Simulator.....</b>	<b>9</b>
Professional Training Tutorials .....	9
Attractive Product Demos.....	9
Animated On-Screen Presentations.....	9
<b>Contact Information .....</b>	<b>10</b>

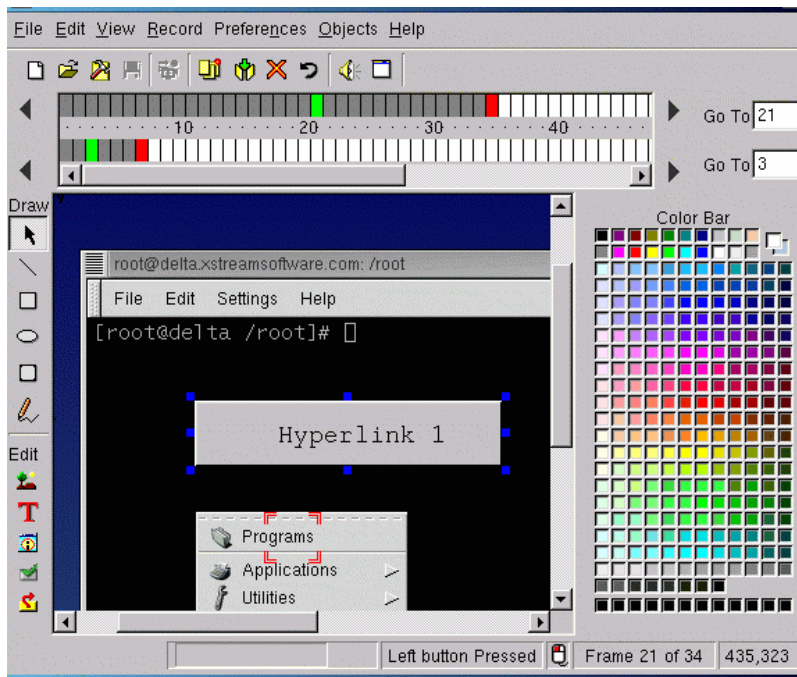
## Introduction to Linux

**Linux** is a freely distributable, open-source operating system. As an implementation of the POSIX specification with additions from both System V and BSD, Linux looks and runs like UNIX, but it was built from an entirely different source code base. Linux has recently emerged as a viable enterprise platform. It is available under the GNU General Public License (GPL), meaning that its source code is open and freely available to all without commercial constraint as long as the individual or company making use of it also abides by the GPL.

One feature that makes Linux so compelling is that it is developed and maintained by a community of software developers around the world. These developers thrive on the concept of producing a stable and powerful alternative to expensive, closed operating systems. Rather than trying to hide problems, Linux developers enjoy discussing them and then teaming up to implement solutions as quickly as possible. System problems, security issues, and future enhancements are all discussed openly. The entire Linux operating system, including the suite of tools, utilities, and applications along with the complete source code, is constantly being debugged, enhanced, and improved. To many people, this represents a level of security and stability unmatched by any commercial operating system.

## What is Linux Simulator?

XStream Software's **Linux Simulator v1.0** is a unique, **100% Programming-free** Simulation-Authoring Technology for Linux. The unique technology of Linux Simulator enables developers to create, customize, and deploy Multimedia Linux Simulations, Presentations, Demos, Tutorials, and other types of content files that realistically duplicate the functionality of the live Linux environment.



**Figure 1: Linux Simulator Interface**

Linux Simulator uses a **continuous** “pixel-based” desktop capturing technique. Rather than taking a snapshot of each screen within the Linux application, Linux Simulator captures pixel changes from screen to screen during the live recording (similar to a video camcorder). This enables Linux Simulator to record and completely capture all live Linux desktop activity and live interactions with Linux-based software applications (including screen changes, mouse movements, mouse clicks, keyboard entries, dialog box

interactivity, etc.). Linux Simulator's unique capturing capability enables developers to create interactive Linux simulations easily (i.e., with reduced time and effort).

Linux Simulator enables developers to create Linux simulations that are **interactive** (where the end user directly interacts with the simulation by clicking the mouse and pressing keys) or **non-interactive** (where the simulation plays back as a hands-free "movie" that the end user simply watches without participating).

In addition to automatically capturing Linux desktop activity (to create interactive or non-interactive Linux simulations), developers may also manually append/insert non-recorded content screens into a simulation file. This capability is ideal for creating presentations, tutorials, and more. Developers have the flexibility to create simulation files containing only captured desktop screens, files containing only inserted/appended content screens, and files containing a combination of captured screens and content screens.

Developers may also customize and enhance simulations using Linux Simulator's drag-and-drop object palettes and other editing options. Developers may incorporate graphics, text, hyperlinks, validation text fields, mouse pointer animations, and more.

Once recording and editing are completed, the finished simulation is generated and compressed in a proprietary .RBX file format. Saved .RBX files are then exported to **Flash (SWF)** format for playback. Generated Flash files are played using the Macromedia Flash Player available from Macromedia's web site ([www.macromedia.com](http://www.macromedia.com)) or through a web browser using the Flash Plug-in (which is already built in within most browsers today)

## **Benefits of Linux Simulator's Advanced Technology**

### **True Simplicity**

Linux Simulator combines the power of a rich multimedia-authoring environment with the simplicity of a desktop-capturing tool. The lack of programming makes creating any type of interactive Linux simulation as easy as clicking a button. With Linux Simulator, you can capture live sessions of screen activity and user actions in real time. Using Linux Simulator's variety of editing tools, a recorded session can be edited and enhanced with the latest in advanced multimedia and interactivity.

### **At Last! No More Programming**

The biggest advantage of Linux Simulator's technology is that it totally eliminates the need for programming or coding of any kind. Linux Simulator handles the difficult work by building interactive simulations automatically. For the first time ever, any Linux application simulation can be created on demand, without the burden of complex programming.

### **Increased Productivity and Cost-Effectiveness**

With Linux Simulator, you can create superior Linux simulations, tutorials, demos, and presentations quickly and inexpensively. Freed from the restraints of time-consuming programming, developers can produce polished simulations on demand. The high speed of Linux Simulator allows fast and efficient production of context-specific simulations that may be edited on the fly.

In addition, the use of Linux Simulator eliminates the need for highly paid programmers and designers, along with the software tools and other facilities that are required for their tasks. By greatly decreasing the amount of manpower and equipment required, Linux Simulator significantly shrinks the costs associated with development.

## Powerful Features of Linux Simulator

Linux Simulator incorporates many unique features that stress power, speed, and particularly flexibility. It can be effectively utilized by general, non-technical users to develop and run a wide variety of sophisticated Linux simulations.

### Create Linux Simulation Files

Linux Simulator allows you to instantly create interactive or non-interactive simulation (.RBX) files by continuously capturing natural, real-time interactions with Linux-based software applications. All screen images and user actions, including mouse pointer movements, mouse clicks, and keystrokes, are captured. Recorded simulations may be interactive (where the end user directly interacts with the simulation by clicking the mouse and pressing keys) or non-interactive (where the simulation plays back as a “movie” that the end user simply watches without participating).

- **Captured Mouse Moves** - As it records the live desktop, Linux Simulator captures mouse position information. As the developer drags the mouse pointer around the screen, Linux Simulator constantly records the coordinates of the mouse pointer at short intervals in order to capture its movement. During simulation playback, all existing mouse position information is retrieved. On each relevant frame, a mouse pointer image is superimposed onto the exact location specified by the recorded mouse coordinates. As a result, the end user sees the same mouse pointer movement that was recorded during desktop capturing.
- **Captured Mouse Clicks** - As it records the live desktop, Linux Simulator captures all left, right, and double mouse clicks made by the user. During simulation playback, captured mouse clicks are validated by the user. Whenever a frame containing a captured mouse click is displayed on the screen, playback pauses for user intervention. In order to make playback proceed, the user must validate the captured click by performing the same action. (For example, assume that a frame contains a captured left-click in the lower left corner of the screen. When the frame appears on screen during playback, the user must left-click the lower left corner of the screen in order to make playback proceed).
- **Captured Keystrokes** - As it records the live desktop, Linux Simulator captures virtually all keystrokes. During simulation playback, captured keystrokes must be validated in the same way as captured mouse clicks. Whenever a frame containing a captured keystroke is displayed on the screen, playback pauses. In order to make playback proceed, the user must press the same key.

In addition to (or instead of) automatically capturing Linux desktop activity, developers may also manually append/insert non-recorded content screens into a simulation file.

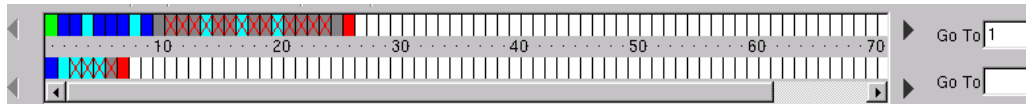
In the Linux Simulator graphical interface, frames can be browsed using the **Navigation Bar**. Each frame can be edited separately by inserting, copying, cutting, pasting, deleting, and purging them. After editing, the file is saved, compressed, and exported to Flash (SWF) format. It may then be distributed to customers for playback. Changes to an existing file are made by loading it into Linux Simulator, performing the necessary changes on any frame, and generating the file. During playback, the captured screens are seamlessly merged with the user action to synthesize an actual session of a live program. The playback is extremely realistic and very convincing, but it consists totally of frame images.

### Powerful Compression

One of Linux Simulator's core strengths is its powerful proprietary compression algorithms. Objects from the Edit and Draw palettes achieve a compression rate of approximately 80%. This allows for richly authored frames without the worry of greatly increased file size.

## Navigation Bar

The **Navigation Bar** allows the developer to edit and navigate through the frames (i.e., the individual screens) of a file during the editing stage.



**Figure 2: Navigation Bar**

Frame editing can be performed, including selecting, cutting, copying, pasting, deleting and undeleting frames. The **Navigation Bar** makes it simple to manage all the frames in a session with a few clicks of the mouse or presses of keyboard keys.

- The **Navigation Bar** features two **Frame Bars**: **All Frames** and **Frames With User Action**. The developer simultaneously views a bar with colored rectangles representing all frames in the session and a bar with colored rectangles representing only the frames that contain captured mouse clicks/keystrokes. This makes it much easier when the developer wishes to navigate/edit only those frames containing mouse clicks or keystrokes.
- There are multiple methods of navigating frames, including clicking a rectangle representing the desired frame, clicking the arrow buttons or the horizontal slider on the **Navigation Bar**, typing a frame number in the Go To box and pressing <ENTER>, or using the keyboard by pressing <CTRL> + Left Arrow or Right Arrow.
- The **Navigation Bar** features selector nodes that can be clicked and dragged to select a range of frames. The keyboard can also be used for this task by pressing <ALT> + Left Arrow or Right Arrow. Once frames are selected, they can be cut, copied, pasted, deleted and undeleted.
- When a frame is deleted, it is marked with a red “X” on the **Navigation Bar** rather than being immediately deleted from the file. This allows the developer the option of undeleting the marked frame if he/she desires. If the developer wishes to delete the frame permanently, Linux Simulator has an option for purging deleted frames.
- Captured mouse clicks and keystrokes may be disabled on a selected range of frames on the **Navigation Bar**.

## Frames

A file created with Linux Simulator is composed of a series of screens known as frames. In addition to captured frames, the developer can insert/append non-recorded content frames. The developer can make various adjustments and enhancements to a frame, including the following:

- Frames may be referenced by number or name. They may be renamed and used as quick references in “go to” objects such as Hyperlinks. For example, instead of typing “16” as a frame reference, the developer may enter the name of the frame (e.g., “intro1” or “main menu”).
- The color of each content frame may be customized.
- Captured user actions (e.g., mouse clicks or keystrokes) on a captured frame can be enabled or disabled.

## Edit Objects

Linux Simulator includes a palette of **Edit** objects that may be used to enhance simulation content. They are as follows:

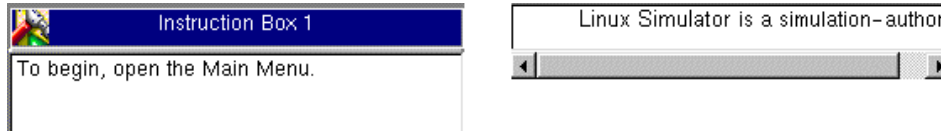
### **Mouse Plot**

The **Mouse Plot** allows the developer to plot mouse positions on a frame in order to create smooth mouse pointer animations. It is ideal for creating demonstrations or instructional simulations in which the end user simply sits back and watches the online animation.

The **Mouse Plot** includes hyperlinking capabilities. The developer may specify any frame within the file to navigate to once the mouse animation has finished.

### **Instruction Box and Text Field**

The **Instruction Box** and **Text Field** are both used to display text in a file. The **Instruction Box** is a resizable window that displays text, while the **Text Field** allows the developer to add text directly to frames.



**Figure 3: Instruction Box and Text Field Objects**

The **Instruction Box** is designed to display extraneous text (such as text-based instructions or technical information), which is not part of the frame content. It is ideal for displaying how-to instructions or steps for tutorials and training materials. The **Text Field** is ideal for displaying any type of text in all areas of a file. Text may be entered in a **Text Field** directly or dynamically loaded to the **Text Field** from an external text file.

### **Bitmap**

The **Bitmap** component allows the developer to assign a custom image file to be displayed on a frame. An image can be aligned, stretched, and positioned in a frame. Linux Simulator supports .BMP, .GIF, and .JPG image files.

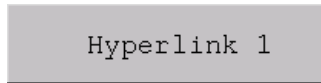
### **Validation Field**

The **Validation Field** is an active field that validates text typed by the end user. Once the end user enters the text and presses a designated action key, the entered text is compared to existing validation strings that have been specified by the developer during editing. If there is a match, the entered text is validated as correct, and any designated events are initiated. If there is no match, the entered text is validated as incorrect (meaning any specified error message is displayed).

- The developer may set the **Validation Field** for string validation by specifying one or more text strings and then associating one or more action keys with each text string.
- A **Validation Field** can be configured to function as a password field in which the end user's typed text appears as asterisk characters (e.g., \*\*\*\*\*).

## Hyperlink

The **Hyperlink** object allows the creation of links between the frames/screens in the file. This provides a means of controlling navigation from screen to screen during simulation playback.



**Figure 4: Hyperlink Object**

- A **Hyperlink** may be configured to link to any other frame in the file.
- The developer may customize the appearance and style of the **Hyperlink** (including choosing whether the **Hyperlink** appears as a push button, radio button, or check box).

## Draw Objects

Linux Simulator includes a palette of **Draw** objects that allow the developer to enhance frames with different lines and shapes. They are as follows:

- **Line** objects can be used to connect text fields, point to frame features, create tables and diagrams, and more. A line can be styled as an arrow, and its size, color and width can be adjusted. It can also be displayed in different styles, including Solid, Dash and Dot.
- **Round Rectangle**, **Rectangle**, and **Ellipse** objects can be used to create tables or visually enhance a captured or inserted frame. The size, color, border style, and border width of these objects can be adjusted. They can also be set to Opaque or Transparent.
- The **Polyline** tool allows the developer to draw straight lines with angles. The color and thickness of the line can be adjusted.

## Color Palette

The **Color Palette** provides a palette of 256 default colors, and it also includes facilities for defining and adding custom colors. It is used to set the foreground color and background color of appended/inserted frames and **Draw** objects.



**Figure 5: Color Palette**

## **Deploying Simulations as Flash (SWF) Files**

To ready a Linux Simulator (.RBX) file for playback, the developer may easily convert it to **Flash (SWF)** format.

Generated Flash (SWF) files may be played using the **Macromedia Flash Player** available from Macromedia's web site ([www.macromedia.com](http://www.macromedia.com)) or through a web browser using the Flash Plug-in (which is already built in within most browsers today).

Flash files are compressed to decrease their size significantly. A typical generated Flash file will be compressed to approximately the same size as an .RBX file, which is a considerable reduction in file size.

## **The Superior Results of Linux Simulator**

With the most advanced technology and innovative features ever seen in a software tool of its kind, Linux Simulator can effectively meet the needs of everyone from CBT developers to technical trainers. For anyone who requires interactive, multimedia content quickly and simply, Linux Simulator is the ultimate solution. The following are just a few examples of the types of powerful Linux simulations that can be developed with Linux Simulator:

### **Professional Training Tutorials**

Developers can use Linux Simulator to create highly professional, straightforward tutorials for different Linux-based software programs and technical procedures. With the ability to incorporate text, graphics, jump buttons, instruction boxes, and more, it's easy to create visually appealing, user-friendly content. Without any programming, developers in any organization can build quick, on-demand training simulations to familiarize employees with new Linux-based software and other high-tech systems. In addition, Linux Simulator can be used to produce more detailed and extensive material for training on Linux-based software products. For the makers of computer-based training (CBT) software, Linux Simulator is an ideal tool that will allow them to develop their products faster and easier. As the technology of Linux Simulator rids them of the need for programmers, CBT development organizations can decrease their production costs significantly.

### **Attractive Product Demos**

Linux Simulator is the perfect Linux demonstration tool for any organization. Demo builders can create multimedia simulations that visually demonstrate the features or functionality of any Linux-based software product with pointer animation, text, and graphics. Companies can use Linux Simulator to produce professional, attractive Linux demos for employees and customers. As a useful supplement to traditional customer support, demos illustrating the solutions to specific technical problems can be created in minutes and distributed to customers on demand.

### **Animated On-Screen Presentations**

Sales and marketing teams can create visually appealing, dynamic Linux presentations for customers and corporate clients, complete with graphics and mouse pointer animations. Using this technology, companies can produce impressive, convincing product presentations and press packages that highlight the features and benefits of their Linux-based software products. Trainers can supplement conventional training material with virtual libraries of movies that allow students to view on-screen functions and procedures in real time. Computer troubleshooters can create movies showing how to address specific errors or problems. In addition, a troubleshooter who requires help in identifying the source of a technical error can use Linux Simulator to create a movie of screen activity before and after the occurrence of the

error. The movie can then be examined by other professionals in the organization or even distributed to outside experts for further analysis.

## **Contact Information**

For more information on **Linux Simulator v1.0**, please contact our sales team either by phone at (613) 731-9443 or by e-mail at [salesupport@xstreamsoftware.com](mailto:salesupport@xstreamsoftware.com).

**XStream Software Inc.**  
2280 St. Laurent Blvd., Suite 200  
Ottawa, Ontario, CANADA  
K1G 4K1

**[www.xstreamsoftware.com](http://www.xstreamsoftware.com)**

info@xstreamsoftware.com (Information)  
salesupport@xstreamsoftware.com (Sales)  
techsupport@xstreamsoftware.com (Support)

Telephone: (613) 731-9443  
Fax: (613) 731-9615