

# What is Entire Screen Builder?

Entire Screen Builder provides a complete range of host terminal communications in a client-server environment.

The central server provides access to applications running on OS/390, VSE/ESA, AS/400, BS2000, UNIX and OpenVMS. Presentation clients (viewers) are available for different styles of presentation, ranging from traditional green screen terminal emulation to "GUIified" Windows dialogs.

The legacy applications are automatically integrated in an Intranet environment without any effort, by using a browser as front-end. HTTP tunneling also allows to use the viewers in the Internet.

The GUI Version provides an easy to implement face-lift for character-mode host applications. Using the transformation rules defined with the Entire Screen Builder SDK, it transforms your legacy applications running on OS/390, VSE/ESA, AS/400, BS2000, UNIX and OpenVMS to graphical user interfaces (GUIs). The following applies to mainframe applications: transformation rules can be defined for applications written in any language.

**Important:**

On UNIX and OpenVMS, the GUI Version can only be used with Natural applications. With the GUI Version, it is not possible to access other UNIX applications or the UNIX operating system.

Many older applications have character-based user interfaces. Reengineering these interfaces to make them more user-friendly would extend the life of the applications, but it can require significant effort. Developers must know how the old interfaces are implemented, and what constraints the new interface must respect to remain compatible with the rest of the application. Using the Entire Screen Builder SDK, you can make the level of effort acceptable.

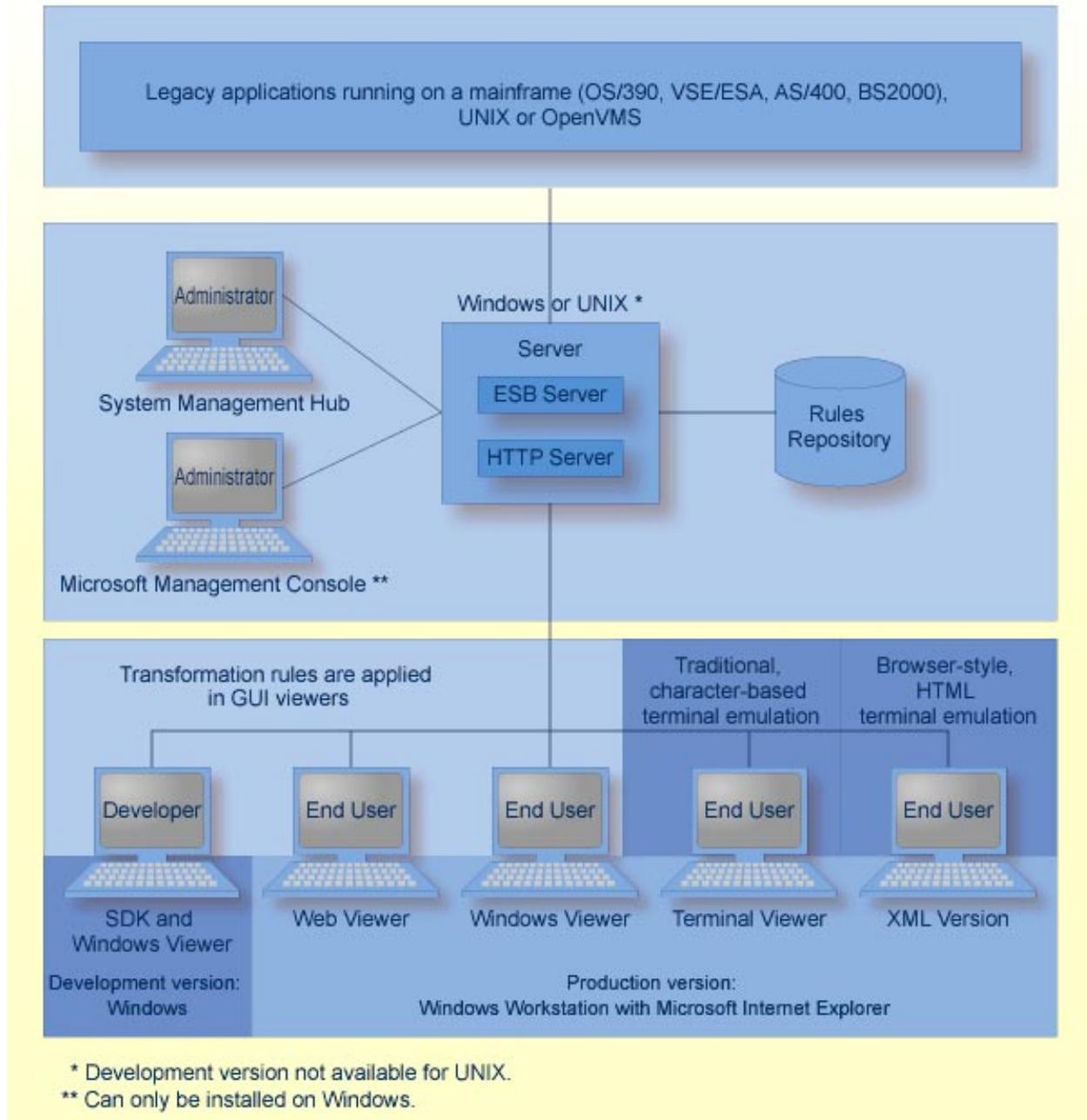
The Terminal Version provides traditional green screen terminal emulation.

The XML Version converts host screens to XML and HTML, allowing zero client footprint host access across a wide range of internet browsers.

**Important:**

On UNIX and OpenVMS, the XML Version can only be used with Natural applications. With the XML Version, it is not possible to access other UNIX applications or the UNIX operating system.

Entire Screen Builder's general architecture is depicted in the following graphic:



Entire Screen Builder consists of the following components:

- SDK (Software Development Kit)
  - Server
  - System Management Hub
  - Server Management
  - Presentation Clients (Viewers)
- 

## SDK (Software Development Kit)

This component is used by the developer to define the transformation rules for the GUI Version. These rules are later used by the server for transforming the character screens to Windows dialogs.

The rules are first created in the developer's local repository. The developer then has to copy the rules to the production environment to make them available to all users.

For detailed information, see the documentation *Defining the Rules Using the SDK*.

## Server

The Entire Screen Builder Server handles both the communication with the host and with the viewers on the client workstations. Multiple and different types of viewers can connect to the server at the same time, each having one connection to a host system.

The Entire Screen Builder Server can be installed in one of the following environments:

- **Windows**  
The Entire Screen Builder Server is automatically installed as a service on one of the supported Windows operating systems (see *Prerequisites* in the *Installation and Configuration* documentation). Configuration information is stored in the Windows registry and in the file *Nswconfig.xml*.
- **UNIX**  
The Entire Screen Builder Server is installed as a normal UNIX process which does not require any special permissions or privileges. Configuration information is stored in the files *ewvreg.reg* and *nswconfig.xml*.

The Entire Screen Builder Server supports scripting for the host sessions. The script language which is used is the core JavaScript version 1.2 plus an Entire Screen Builder specific object which enables you to interact with the host session. See also: *Scripting, User Exits and APIs*.

## System Management Hub

The administrator can use the System Management Hub (which is the standard, GUI-based, central point of administration for Software AG products) to administrate the Entire Screen Builder Server.

The System Management Hub is used to start and stop the Entire Screen Builder Server, to define and administrate host sessions as well as user and group profiles, to load and unload modules, to send messages to the clients currently connected to the server, etc.

For detailed information, see Entire Screen Builder's *System Management Hub* documentation.

## Server Management

The Entire Screen Builder Server Management tool is a Microsoft Management Console snap-in which offers administration and monitoring facilities for the Entire Screen Builder Server. It allows the administrator to show the running connections and how the number of connections to the server varies with time. It can also be used to define and administrate host sessions, key schemes and color schemes. Host sessions can be added and newly added sessions can be changed without having to stop the Entire Screen Builder Server.

The Entire Screen Builder Server Management tool is intended to be used when the Entire Screen Builder Server is running, as a complement to the System Management Hub which has administration facilities but requires the Entire Screen Builder Server to be stopped for many functions.

For detailed information, see the *Server Management* documentation.

## Presentation Clients (Viewers)

A presentation client is used by the end-user. Different presentation clients are available with the different Entire Screen Builder products.

### GUI Version

The Entire Screen Builder GUI Version contains the GUI viewers in which the transformation rules are applied. The following viewers are available:

- Web Viewer running in a browser.
- Windows Viewer running as a separate Windows application.

Each character screen that the user invokes is transformed to a Windows dialog according to the transformation rules defined by the developer. Fields are thus converted to simple GUI controls such as edit boxes, option (radio) buttons or check boxes. The legacy applications work in the same way as before, without any modification or recompilation.

For further information, see the *GUI Viewers* documentation.

### Terminal Version

The Entire Screen Builder Terminal Version contains the following viewers:

- Terminal Viewer running in a browser.
- Standalone Terminal Viewer running as a separate Windows application.

Transformation rules are not applied in the Terminal Viewer. It always shows the screen as sent from the host.

For further information, see the *Terminal Viewer* documentation.

### XML Version

The Entire Screen Builder XML Version does not contain a special viewer. A standard internet browser (for example, Microsoft Internet Explorer 5 or 6) is used to access the host screens. The server output module of the XML Version converts the host screens to XML. In an HTTP server extension, the XML data stream is then translated to an HTML page using an XSL style sheet. Generic style sheets provided by Software AG or customer-supplied style sheets can be used.

The Entire Screen Builder Web Viewer and the Terminal Viewer are implemented as Microsoft ActiveX controls. Both have a COM interface that offers properties, methods and events which enable you to integrate the viewers into your Windows application and control the behavior of the ActiveX control. You can use the API from a wide variety of languages: C++, JavaScript, Visual Basic, etc. See also: *Scripting, User Exits and APIs*.

## Terminal Model Support

The following table shows the terminal model support for the different presentation clients:

	<b>Model 2 (24 x 80)</b>	<b>Model 3 (32 x 80)</b>	<b>Model 4 (43 x 80)</b>	<b>Model 5 (27 x 132)</b>	<b>UNIX</b>
<b>GUI Viewers</b>	yes	no	no	no	24 x 80
<b>Terminal Viewer</b>	yes	yes	yes	yes	24 x 80
<b>XML Version</b>	yes	yes	yes	yes	24 x 80