The DOM Level 2 Event Model is designed with two main goals.

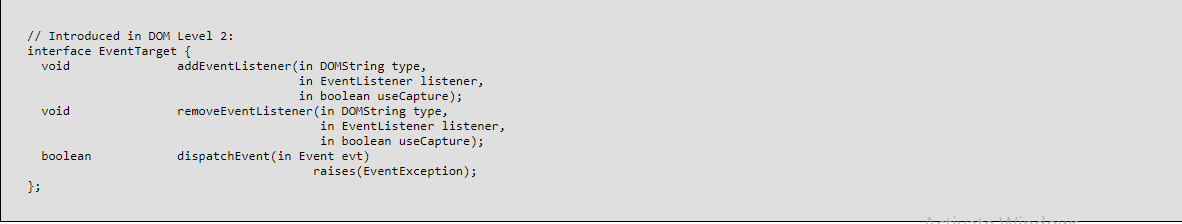
The first goal is the design of a generic event system which allows registration of event handlers, describes event flow through a tree structure, and provides basic contextual information for each event. Additionally, the specification will provide standard modules of events for user interface control and document mutation notifications, including defined contextual information for each of these event modules.

The second goal of the event model is to provide a common subset of the current event systems used in [DOM Level 0](https://www.w3.org/TR/DOM-Level-2-Events/glossary.html#dt-DOM-Level-0) browsers. This is intended to foster interoperability of existing scripts and content. It is not expected that this goal will be met with full backwards compatibility. However, the specification attempts to achieve this when possible.

The following sections of the Event Model specification define both the specification for the DOM Event Model and a number of conformant event modules designed for use within the model. The Event Model consists of the two sections on event propagation and event listener registration and the Event interface.

A DOM application may use the hasFeature(feature, version) method of the DOMImplementation interface with parameter values "Events" and "2.0" (respectively) to determine whether or not the event module is supported by the implementation. In order to fully support this module, an implementation must also support the "Core" feature defined in the DOM Level 2 Core specification.

The EventTarget interface is implemented by all Nodes in an implementation which supports the DOM Event Model. Therefore, this interface can be obtained by using binding-specific casting methods on an instance of the Node interface. The interface allows registration and removal of **[EventListeners](https://www.w3.org/TR/DOM-Level-2-Events/events.html" \l "Events-EventListener)** on an EventTarget and dispatch of events to that EventTarget.



**Interface*****EventListener* (introduced in DOM Level 2)**

The EventListener interface is the primary method for handling events. Users implement the EventListener interface and register their listener on an [EventTarget](https://www.w3.org/TR/DOM-Level-2-Events/events.html" \l "Events-EventTarget) using the AddEventListener method. The users should also remove their EventListener from its [EventTarget](https://www.w3.org/TR/DOM-Level-2-Events/events.html" \l "Events-EventTarget) after they have completed using the listener.

When a Node is copied using the cloneNode method the EventListeners attached to the source Node are not attached to the copied Node. If the user wishes the same EventListeners to be added to the newly created copy the user must add them manually.

**IDL Definition**

// Introduced in DOM Level 2:

interface EventListener {

void handleEvent(in Event evt);

};

**Interface*****Event* (introduced in DOM Level 2)**

The Event interface is used to provide contextual information about an event to the handler processing the event. An object which implements the Event interface is generally passed as the first parameter to an event handler. More specific context information is passed to event handlers by deriving additional interfaces from Event which contain information directly relating to the type of event they accompany. These derived interfaces are also implemented by the object passed to the event listener.

