# Glossary

**accelerator**

Any efficient, alternate method of implementing a series of commands. Specifically, drag-and-drop is an accelerator because it replaces one or more commands that could be executed through a series of command line scripts or menu selections.

**additional action**

Any action that takes place after the completion of a drag-and-drop data transfer. This includes actions that comprise normal termination of the conversation. Addenda (or "side effects") are included in this category.

**adjust mouse button**

The mouse button (the center one, by default) that is used to adjust (add or remove) selections.

**alternate transport medium**

(ATM) Any communication channel other than the X wire; for instance, the ToolTalk service and sockets.

**anchor**

A connection point in a data file which supports one end of a link.

**animation**

There are two primary types of animation, tracking and previewing.

**animation, previewing**

A visual indication of receptivity to a drop. This may be indicated by a pointer change, for instance.

**animation, tracking**

The process of making an image of the object move across the desktop in synchronization with the user's mouse movement.

**API**

application programming interface.

**Classing Engine**

A mechanism that permits an application to query a database (the Classing Engine database) to determine the attributes of a desktop object.

**conversation**

The negotiations necessary to determine the format, transport method, and other considerations pertinent to the data to be transferred in a drag-and-drop operation.

**data span**

A segment of on-screen data. It can be a segment of text, digitized audio, video, and so forth.

**desktop object**

A discrete on-screen representation of data. This could be a data span, an application icon, file glyph, etc.

**drag and drop**

The overall concept of using a mouse to select a desktop object and move or copy it to another desktop object.

**Drop Site Database Manager (DSDM)**

A process (not the sending client) responsible for maintaining a registry or database of potential drop sites for drag-and-drop operations.

**File type**

Refers to a file's format (e.g., ASCII, PostScript, and Sun raster files), its parent application (FrameMaker, Lotus 1-2-3 data files) or the application executable itself (File Manager, Mail Tool, or Wingz® executable file).

**gesture, drag and drop**

Holding down the selection button over an existing selection, moving the mouse, then releasing the selection button. This causes a representation of the selection to move to the point where the button is released.

**glyph**

Any graphical element on the desktop. A glyph may be a button, a folder, or other graphical element representing a document or file.

**hints**

Suggestions that a source can provide to a destination concerning possible ways to deal with the data exchange.

**icon**

A closed representation of a window. An window displayed as an icon is still running. An icon may change in appearance to show the state of the application; for instance, the familiar Mail Tool icon shows whether new mail has arrived by displaying envelopes in a tray.

**interplay**

The ability of various objects on the OpenWindows desktop to exchange data without requiring user intervention.

**match attributes**

Each entry in a namespace table will have one or more match attribute values, which will be used by the namespace manager in matching client arguments.

**menu mouse button**

The mouse button (with a three-button mouse, by default the right one) that is used to display a menu associated with a desktop object.

**namespace**

Refers to the space from which an object name is derived and understood. Files are named within the file namespace, printers are named within the printer namespace. You cannot name a printer by using a file name.

**namespace table**

A namespace table is the place where all namespace information is stored, for use by the CE as well as a namespace manager. Each namespace table consists of entries (rows) and each entry consists of a set of named attributes.

**namespace manager**

Every namespace has a namespace manager. A namespace manager is a piece of code that performs the matching of client supplied arguments with the attributes of entries in a table.

Different namespace managers will use different matching logic. Consider two examples - (1) mapping file information to a file type and (2) mapping a type name to type attributes.

In case (1), the namespace manager for files might have to know a file name, a magic number and fstat() information in order to match with its own match attributes of file name pattern, magic number and an fstat() mask.

In case (2), all the type namespace manager has to know is the name of a type, which it will exact match with its type name match attribute.

**Notice**

A special pop-up window initiated by an application. A Notice warns of errors or potential loss of data. The application will not accept further input until the user dismisses the Notice by selecting a desired action.

**object**

See desktop object.

**object type derivation**

The process by which an object's name and content information is examined to determine the type of an object. This is not the same process as the one used to give an object its type.

**OLIT**

The OPEN LOOK Intrinsics Toolkit.

**rendezvous**

The set of events that are necessary to identify the sending client to the receiving client.

**select**

To distinguish an object (or objects) on the desktop so they may be operated on.

**select mouse button**

The mouse button (the left one, by default) that is used to select objects, set the insert point, drag objects, and set/reset buttons.

**ToolTalk**

A service for communications between applications on the desktop.

**transport**

The means by which an object is passed from one process to another.

**type-specific attributes**

Each object type has a set of attributes associated with it e.g. its methods, icons etc. These attributes are referred to as type-specific attributes.

**type database**

The database used to map an object type to its attributes. In the context of the CE, a type database is a namespace table, where types are named and their attributes stored.

**workspace**

The background area of a display screen on which windows and icons are displayed.