

This chapter discusses sequence grabber panel components. Sequence grabber components create a settings dialog box that includes items that are managed by sequence grabber panel components and sequence grabber channel components. **Sequence grabber panel components** allow sequence grabber components to obtain configuration information from the user for a particular sequence grabber channel component. Applications never call sequence grabber panel components directly; application developers use panel components only by calling the sequence grabber component.

This chapter is divided into the following sections:

- “About Sequence Grabber Panel Components” provides a general introduction to components of this type.
- “Creating Sequence Grabber Panel Components” discusses how sequence grabbers use these components.
- “Sequence Grabber Panel Components Reference” presents detailed information about the functions that are supported by these components.
- “Summary of Sequence Grabber Panel Components” contains a condensed listing of the constants and functions supported by these components.

This chapter addresses developers of sequence grabber panel components. If you plan to create a sequence grabber panel component, you should read the entire chapter. If you are writing an application that uses components of this type, you do not need to read this chapter. Refer to the chapter “Sequence Grabber Components” in this book for information about sequence grabber components and how to display the settings dialog box to the user.

As components, sequence grabber panel components rely on the facilities of the Component Manager. In order to use any component, your application must also use the Component Manager. If you are not familiar with this manager, see the chapter “Component Manager” in *Inside Macintosh: More Macintosh Toolbox*. In addition, you should be familiar with sequence grabber components and sequence grabber channel components. See the chapters “Sequence Grabber Components” and “Sequence Grabber Channel Components” in this book for more information.

#### Note

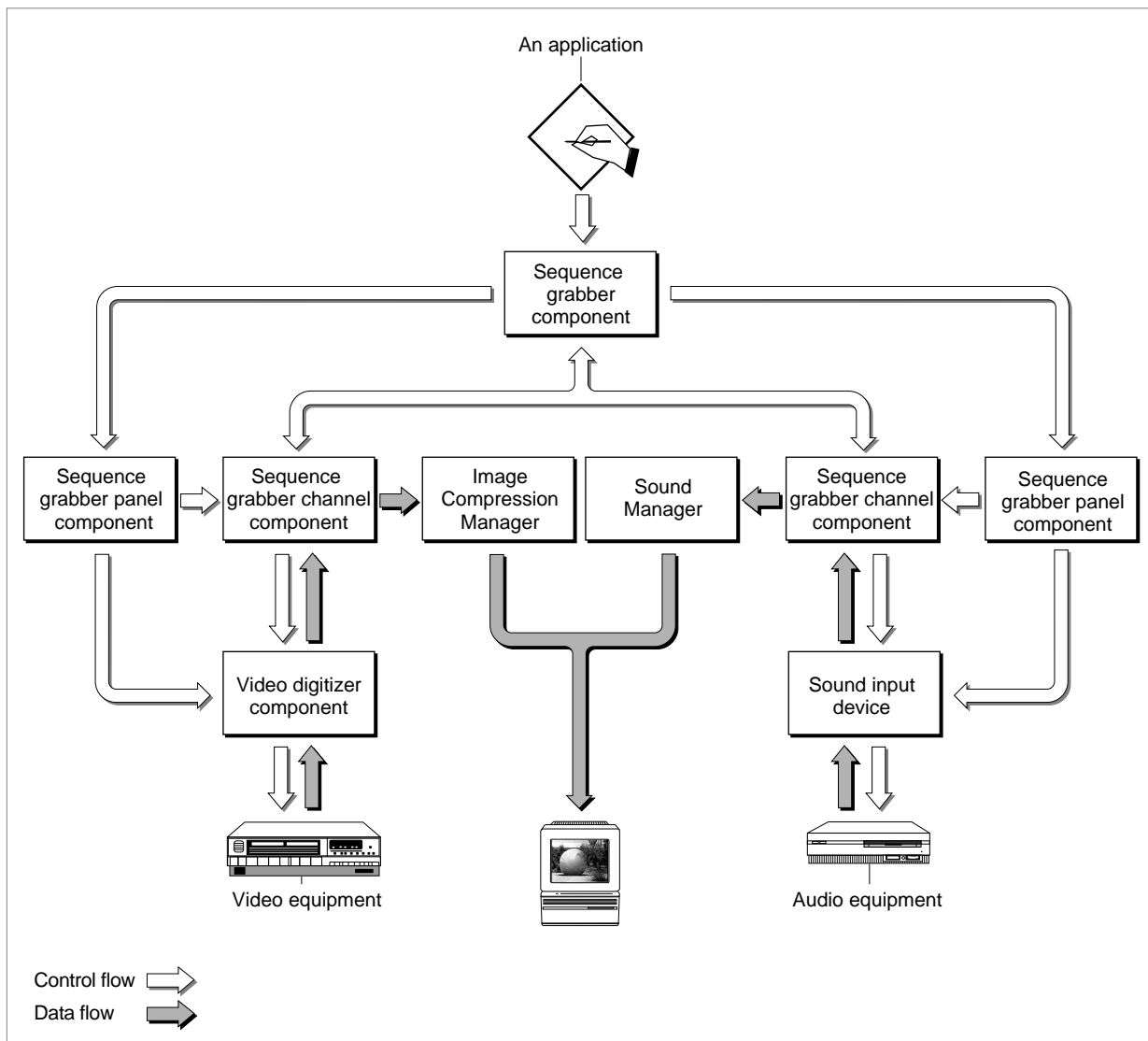
The text in this chapter makes numerous references to sequence grabber components, sequence grabber channel components, and sequence grabber panel components. For the sake of brevity, shortened names have been adopted for each of these components. Consequently, you will often find sequence grabber components referred to as *sequence grabbers*; sequence grabber channel components as *channel components*; and sequence grabber panel components as *panel components*. ♦

## About Sequence Grabber Panel Components

---

This section provides background information about sequence grabber panel components. After reading this section, you should understand why these components exist and whether you need to create one.

Sequence grabber panel components augment the capabilities of sequence grabber components and sequence grabber channel components by allowing sequence grabbers to obtain configuration information from the user for a particular digitizing source that is managed by a channel component. Consequently, sequence grabbers, channel components, and panel components have a close relationship. Figure 7-1 shows this relationship and how these components interact with one another to place digitized data into a QuickTime movie.

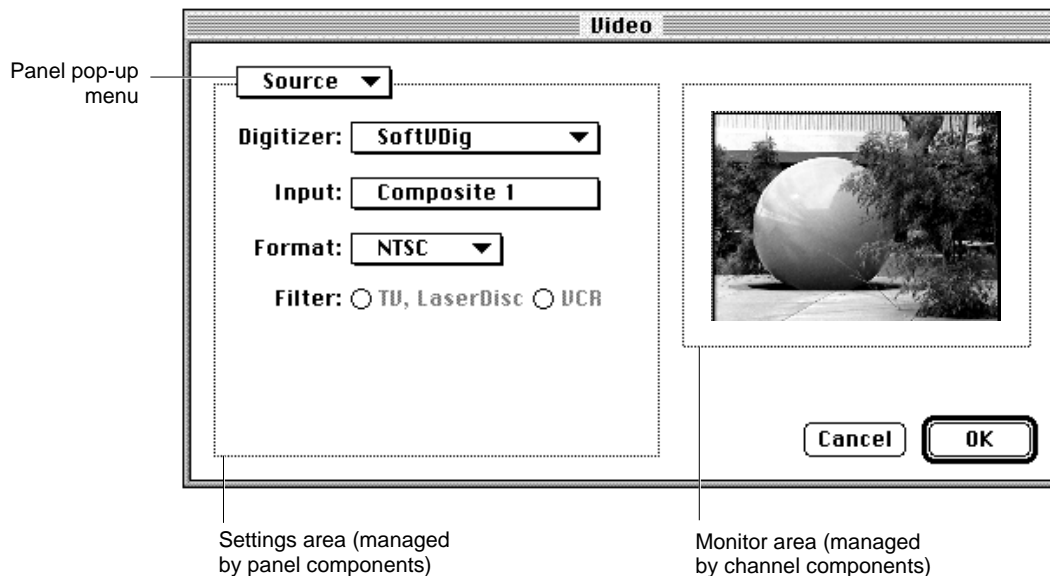
**Figure 7-1** Sequence grabbers, channel components, and panel components

Sequence grabbers present a settings dialog box to the user whenever an application calls the `SGSettingsDialog` function (see the chapter “Sequence Grabber Components” in this book for more information about this sequence grabber function). Applications never call sequence grabber panel components directly; application developers use panel components only by calling the sequence grabber component.

## Sequence Grabber Panel Components

Although the sequence grabber creates the dialog box and manages its interactions with the user, portions of the dialog box are controlled by panel components and channel components. Figure 7-2 shows a sample dialog box and identifies the various parts of the dialog box.

**Figure 7-2** A sample sequence grabber settings dialog box



The sequence grabber creates the dialog box itself and manages the OK and Cancel buttons and the panel pop-up menu. Channel components are responsible for the monitor area on the right side of the dialog box. Panel components manage the settings area immediately below the panel pop-up menu. Only one panel component is active at any given time; the user selects a panel component by manipulating the panel pop-up menu.

When the user selects a specific panel component, the sequence grabber works with that component to build the panel settings dialog area and present it to the user. The panel component processes dialog events and mouse clicks as appropriate and validates the user's settings. The sequence grabber then retrieves the settings from the panel component and stores those settings.

There are two circumstances under which you should consider creating a sequence grabber panel component: first, if you want to support special digitizing equipment in the QuickTime environment; and, second, if you have created your own sequence grabber channel component.

If you have created special digitizing equipment, you may not have to create a special channel component for your equipment—the channel components provided by Apple may be sufficient for your needs. By providing a special panel component, however, you can allow the user to take advantage of your equipment's special capabilities.

If you have created your own channel component, you must create an accompanying panel component to allow the user to configure your channel.

## Creating Sequence Grabber Panel Components

This section discusses how to create a sequence grabber panel component. You should read this section if you are creating a panel component.

Applications do not call panel components directly. Rather, they invoke a sequence grabber's settings dialog box by calling the `SGSettingsDialog` function. In response, the sequence grabber presents the settings dialog box to the user. When the user selects a specific settings panel, the sequence grabber invokes the appropriate panel component.

Panel components provide a number of functions that allow sequence grabbers to manage their relationships with panel components. See “Managing Your Panel Component” beginning on page 7-15 for complete descriptions of these functions.

Panel components are not responsible for saving their settings information. Sequence grabbers manage this information on behalf of panel components, and a sequence grabber may combine configuration information from several panel components in order to build up the complete configuration for an elaborate digitizing environment. Panel components provide functions that allow sequence grabbers to obtain this configuration information. See “Managing Your Panel's Settings” beginning on page 7-24 for more information about these functions.

Sequence grabbers store this configuration data in user data items. The Movie Toolbox provides a number of functions that allow you to create and manage user data items. If you are not familiar with these functions, see the chapter “Movie Toolbox” in *Inside Macintosh: QuickTime* for more information.

Apple has defined a component type value for sequence grabber panel components. You can use the following constant to specify this component type.

```
#define SeqGrabPanelType 'sgpn' /* panel component type */
```

Sequence grabber panel components use their component subtype and manufacturer values to indicate the type of configuration services they provide. The subtype value indicates the media type supported by the panel component. This value should correspond to the component subtype value of channel components that may be configured by the panel component. For example, a panel component that manages video settings would have a subtype of 'vide' (this value is defined by the Movie Toolbox's `VideoMediaType` constant).

## Sequence Grabber Panel Components

The manufacturer field contains a unique identifier for each panel component. The value should indicate something about the specific services provided by the component. For example, Apple has defined the following manufacturer values:

```
#define SeqGrabCompressionPanelType  'sour'  /* input source
                                           selection */
#define SeqGrabSourcePanelType        'cmpr'  /* compression
                                           settings */
```

In general, Apple has reserved all lowercase values of component subtypes and manufacturer codes.

Apple has defined a functional interface for sequence grabber panel components. For information about the functions that your component must support, see “Sequence Grabber Panel Components Reference” beginning on page 7-14. You may use the following constants to refer to the request codes for each of the functions that your component must support:

```
enum {
    /* sequence grabber panel request codes */

    kSGCPanelGetDitlSelect      = 0x200, /* SGPanelGetDITL */
    kSGCPanelCanRunSelect       = 0x202, /* SGPanelCanRun */
    kSGCPanelInstallSelect      = 0x203, /* SGPanelInstall */
    kSGCPanelEventSelect        = 0x204, /* SGPanelEvent */
    kSGCPanelItemSelect         = 0x205, /* SGPanelItem */
    kSGCPanelRemoveSelect       = 0x206, /* SGPanelRemove */
    kSGCPanelSetGrabberSelect    = 0x207, /* SGPanelSetGrabber */
    kSGCPanelSetResFileSelect    = 0x208, /* SGPanelSetResFile */
    kSGCPanelGetSettingsSelect   = 0x209, /* SGPanelGetSettings */
    kSGCPanelSetSettingsSelect   = 0x20A, /* SGPanelSetSettings */
    kSGCPanelValidateInputSelect = 0x20B  /* SGPanelValidateInput */
};
```

Before reading the rest of this chapter, you should know how to create components. See the chapter “Component Manager” in *Inside Macintosh: More Macintosh Toolbox* for a complete discussion of components, how to use them, and how to create them.

The next section contains sample code for the creation of a sequence grabber panel component that acts as a settings dialog box for PICT images. To create a sequence grabber panel component, you set up the global variables and implement the required Component Manager request codes and the functions that are private to your particular component. Then you manage the dialog box and work with the settings in the dialog box.

## Implementing the Required Component Functions

Listing 7-1 supplies the component dispatchers for the sequence grabber panel component together with the required functions for open, close, can do, and version.

**Listing 7-1** Implementing the required functions

```
#define sgcPictShowTicksType 'TICK'

typedef struct {
    ComponentInstance    self;
    ControlHandle        ch;
} PictPanelGlobalsRecord, *PictPanelGlobals;

/* only for PICT channels */
pascal ComponentResult SGSetShowTickCount (SGChannel c,
                                           Boolean show) = {0x2f3c, 2, 0x100, 0x7000, 0xA82A};
pascal ComponentResult SGGetShowTickCount (SGChannel c,
                                           Boolean *show) = {0x2f3c, 4, 0x101, 0x7000, 0xA82A};
pascal ComponentResult PictPanelDispatcher
    (ComponentParameters *params, Handle storage)

{
    OSErr err = badComponentSelector;
    ComponentFunction componentProc = 0;
    switch (params->what) {

        case kComponentOpenSelect:
            componentProc = PictPanelOpen; break;
        case kComponentCloseSelect:
            componentProc = PictPanelClose; break;
        case kComponentCanDoSelect:
            componentProc = PictPanelCanDo; break;
        case kComponentVersionSelect:
            componentProc = PictPanelVersion; break;
        case kSGCPanelGetDitlSelect:
            componentProc = PictPanelPanelGetDitl; break;
        case kSGCPanelInstallSelect:
            componentProc = PictPanelPanelInstall; break;
        case kSGCPanelItemSelect:
            componentProc = PictPanelPanelItem; break;
        case kSGCPanelRemoveSelect:
            componentProc = PictPanelPanelRemove; break;
```

## Sequence Grabber Panel Components

```

        case kSGCPanelGetSettingsSelect:
            componentProc = PictPanelPanelGetSettings; break;
        case kSGCPanelSetSettingsSelect:
            componentProc = PictPanelPanelSetSettings; break;
    }

    if (componentProc)
        err = CallComponentFunctionWithStorage (storage, params,
                                                componentProc);

    return err;
}

pascal ComponentResult PictPanelCanDo (PictPanelGlobals store,
                                       short ftnNumber)
{
    switch (ftnNumber) {
        case kComponentOpenSelect:
        case kComponentCloseSelect:
        case kComponentCanDoSelect:
        case kComponentVersionSelect:
        case kSGCPanelGetDitlSelect:
        case kSGCPanelInstallSelect:
        case kSGCPanelItemSelect:
        case kSGCPanelRemoveSelect:
        case kSGCPanelGetSettingsSelect:
        case kSGCPanelSetSettingsSelect:
            return true;
        default:
            return false;
    }
}

pascal ComponentResult PictPanelVersion (PictPanelGlobals store)
{
    return 0x00020001;
}

```



## Sequence Grabber Panel Components

```

pascal ComponentResult PictPanelOpen (PictPanelGlobals store,
                                      ComponentInstance self)
{
    OSErr err;

    /* allocate global variables */
    store = (PictPanelGlobals) NewPtrClear
        (sizeof(PictPanelGlobalsRecord));
    if (err = MemError()) goto bail;
    SetComponentInstanceStorage (self, (Handle)store);

    /* remember the component instance identification number */
    store->self = self;

bail:
    return err;
}

pascal ComponentResult PictPanelClose (PictPanelGlobals store,
                                       ComponentInstance self)
{
    if (store) DisposePtr ((Ptr)store);
    return noErr;
}

```

## Managing the Dialog Box

This section gives details on the functions that the panel component must provide so that the sequence grabber can load the component's items into the settings dialog box and receive and process dialog events.

1. To prepare to add the component's items to the settings dialog box, the sequence grabber obtains the item list by calling the `SGPanelGetDITL` function (described on page 7-18).
2. Once it has installed the items, the sequence grabber calls the `SGPanelInstall` function (described on page 7-19), which sets up the state of the dialog box (for example, a checkbox) and gives the panel component an opportunity to set initial values.
3. When the panel component is loaded into the settings dialog box and active, it may receive and process dialog events and mouse clicks. The component's `SGPanelEvent` function (described on page 7-22) processes individual dialog events.
4. Whenever the user clicks a dialog item, the sequence grabber calls the `SGPanelItem` function (described on page 7-21).
5. Before the sequence grabber removes the items from the settings dialog box, it calls the `SGPanelRemove` function (described on page 7-20).

## Sequence Grabber Panel Components

Listing 7-2 provides an example of the management of the settings dialog box for a sequence grabber that displays PICT images. The component item displayed in the dialog box in this case is a tick count checkbox.

---

**Listing 7-2** Managing the settings dialog box

```
pascal ComponentResult PictPanelPanelGetDitl
                                (PictPanelGlobals store,
                                 Handle *ditl)
{
    /*
       Get and detach the dialog box template. Note that
       the sequence grabber has already opened the resource file.
    */
    *ditl = GetResource ('DITL', 7001);
    if (!*ditl) return resNotFound;
    DetachResource (*ditl);
    return noErr;
}

pascal ComponentResult PictPanelPanelInstall
                                (PictPanelGlobals store, SGChannel c,
                                 DialogPtr d, short itemOffset)
{
    Rect r;
    short kind;
    Handle h;
    Boolean ticksShowing;

    /* set up the initial state of the checkbox */
    GetDItem (d, 1 + itemOffset, &kind, &h, &r);
    store->ch = (ControlHandle)h;
    SGGetShowTickCount (c, &ticksShowing);
    SetCtlValue (store->ch, ticksShowing);

    return noErr;
}

pascal ComponentResult PictPanelPanelItem
                                (PictPanelGlobals store, SGChannel c,
                                 DialogPtr d, short itemOffset,
                                 short itemNum)
```

## Sequence Grabber Panel Components

```

{
    /* if the item clicked was your checkbox, update its state */
    if ((itemNum - itemOffset) == 1) {
        Boolean showing = GetCtlValue (store->ch);
        SetCtlValue (store->ch, !showing);
        SGSetShowTickCount (c, !showing);
    }

    return noErr;
}

pascal ComponentResult PictPanelPanelRemove
(
    PictPanelGlobals store,
    SGChannel c, DialogPtr d,
    short itemOffset)
{
    /* forget that it ever had a control */
    store->ch = nil;
    return noErr;
}

```

## Managing Your Panel's Settings

To allow the sequence grabber to work with your panel's settings, your panel component must allow the sequence grabber to

- retrieve the panel's current settings by calling your `SGPanelGetSettings` function (described on page 7-24)
- restore those settings to some previous values by using your `SGPanelSetSettings` function (described on page 7-25)

Listing 7-3 gives an example in which the settings are managed in a user list that contains tick count information for a panel component for PICT images.

**Listing 7-3** Managing the settings for a panel component

```

pascal ComponentResult PictPanelPanelGetSettings
(
    PictPanelGlobals store, SGChannel c,
    UserData *result, long flags)
{
    OSErr      err;
    UserData   ud;
    Boolean    ticksShowing;

```

## Sequence Grabber Panel Components

```

    /* create a user data list containing your state */
    if (err = NewUserData (&ud)) goto bail;
    if (err = SGGetShowTickCount (c, &ticksShowing)) goto bail;
    if (err = SetUserDataItem (ud, &ticksShowing,
                              sizeof (ticksShowing),
                              sgcPictShowTicksType, 1)) goto bail;

bail:
    if (err) {
        DisposeUserData(ud);
        ud = 0;
    }
    *result = ud;

    return err;
}

pascal ComponentResult PictPanelPanelSetSettings
    (PictPanelGlobals store, SGChannel c,
     UserData ud, long flags)
{
    Boolean ticksShowing;

    /* restore the state from the specified user data list */
    if (GetUserDataItem (ud, &ticksShowing,
                        sizeof (ticksShowing),
                        sgcPictShowTicksType, 1) == noErr)
        SGSetShowTickCount (c, ticksShowing);

    return noErr;
}

```

## Sequence Grabber Panel Components Reference

---

This section describes the constants and functions that your sequence grabber panel component may support. Some of these functions are optional—your component should support only those functions that are appropriate to it.

## Component Flags for Sequence Grabber Panel Components

The Component Manager allows you to specify information about your component's capabilities in the `componentFlags` field of the component description record. Sequence grabber panel components use the `componentFlags` field to indicate specific information about their capabilities.

The following flags are currently defined:

```
enum {
    channelFlagDontOpenResFile = 2,    /* do not open resource
                                         file */
    channelFlagHasDependency = 4       /* needs special hardware */
};
```

These flags control how sequence grabbers manage their connection with your panel component. The `channelFlagDontOpenResFile` flag instructs the sequence grabber not to open your component's resource file. By default, the sequence grabber opens your component's resource file for you, and then provides you with the appropriate file reference number. In general, this is convenient. However, if your component is linked with your application and does not have its own resource file, you may not want the sequence grabber to try to open the resource file. In such cases, set this flag to 1.

The `channelFlagHasDependency` flag allows you to tell the sequence grabber that your panel component requires special digitizing hardware. If you set this flag to 1, the sequence grabber gives your component an opportunity to verify that it can work in the current hardware environment—by calling your component's `SGPanelCanRun` function (described on page 7-17).

## Functions

This section describes the functions that may be supported by sequence grabber panel components. It is divided into the following topics:

- “Managing Your Panel Component” discusses the functions that allow sequence grabber components to load, configure, and unload your panel component.
- “Processing Your Panel's Events” describes the functions that allow your component to receive and process events in your panel.
- “Managing Your Panel's Settings” tells you about the functions that allow sequence grabber components to collect and reset your panel's settings.

## Managing Your Panel Component

Sequence grabber components load, configure, and unload your panel component. As part of this process, the sequence grabber installs your panel's dialog items into the settings dialog box and may open your component's resource file. Panel components

## Sequence Grabber Panel Components

provide a number of functions that allow the sequence grabber to manage its relationship with panel components. This section discusses those functions.

After opening a connection to your panel component, the sequence grabber identifies itself to your component by calling your `SGPanelSetGrabber` function. The sequence grabber then tries to determine whether your component can work with its associated channel component by calling your `SGPanelCanRun` function. The sequence grabber calls this function only if you have set the `channelFlagHasDependency` component flag to 1.

Once the sequence grabber has determined that your panel component can work with its channel component, the sequence grabber may open your component's resource file (unless you have set the `channelFlagDontOpenResFile` component flag to 1). Once it has opened the resource file, it passes the file's reference number to you by calling your `SGPanelSetResFile` function.

Next, the sequence grabber prepares to add your component's items to the settings dialog box. The sequence grabber obtains your item list by calling your `SGPanelGetDITL` function. Once it has installed the items, it calls your `SGPanelInstall` function, giving you an opportunity to set initial values.

Before the sequence grabber removes your items from the settings dialog box, it calls your `SGPanelRemove` function.

## SGPanelSetGrabber

---

The `SGPanelSetGrabber` function allows a sequence grabber component to identify itself to your panel component. This is typically the first function the sequence grabber component calls after opening your panel component.

```
pascal ComponentResult SGPanelSetGrabber
                                (SeqGrabPanelComponent s,
                                 SeqGrabComponent sg);
```

s	Identifies the sequence grabber component's connection to your panel component.
sg	Identifies a connection to the sequence grabber component that is using your panel component. Your component may use this connection to call sequence grabber component functions.

### DESCRIPTION

A sequence grabber component calls your `SGPanelSetGrabber` function in order to identify itself to your panel component. Your component can use the provided connection to call sequence grabber functions, either to determine the characteristics of the current capture operation or to alter those characteristics.

RESULT CODE

badComponentSelector	0x80008002	Function not supported
----------------------	------------	------------------------

SGPanelCanRun

The SGPanelCanRun function allows a sequence grabber component to determine whether your panel component can work with the current sequence grabber channel component.

```
pascal ComponentResult SGPanelCanRun (SeqGrabPanelComponent s,  
                                       SGChannel c);
```

- |   |   |
|---|---|
| s | Identifies the sequence grabber component's connection to your panel component.   |
| c | Identifies a connection to a sequence grabber channel component. You must determine whether your panel component can operate with this channel component and its associated channel hardware. |

DESCRIPTION

A sequence grabber component calls your SGPanelCanRun function in order to determine whether your component can work with a specified sequence grabber channel component and its associated hardware. If your component works only with certain hardware, you should support this function.

Set the channelFlagHasDependency component flag to 1 to cause the sequence grabber component to call this function.

The sequence grabber component provides you with a connection to the channel component in question. Your component should query the channel component to determine whether you can operate with it. You may want to use channel component functions to determine the characteristics of the digitization source attached to the channel. If your component can work with the specified channel, return a result code of noErr. Otherwise, return an appropriate sequence grabber or sequence grabber channel component result code.

If your panel component can only support a limited number of connections, you should regulate the number of active connections in your SGPanelCanRun function. Return a nonzero result code to indicate to the sequence grabber that your panel component cannot support the current connection.

RESULT CODES

noDeviceForChannel	-9408	Cannot work with specified channel
badComponentSelector	0x80008002	Function not supported
Other appropriate sequence grabber or sequence grabber channel result codes		

## SGPanelSetResFile

---

Unless you instruct it otherwise, the sequence grabber component opens your panel component's resource file for you. The `SGPanelSetResFile` function allows the sequence grabber to pass you the resource file's reference number. The sequence grabber also calls this function when it closes your resource file.

```
pascal ComponentResult SGPanelSetResFile
                                (SeqGrabPanelComponent s,
                                 short resRef);
```

<code>s</code>	Identifies the sequence grabber component's connection to your panel component.
<code>resRef</code>	Contains a reference number that identifies your component's resource file. After it closes your resource file, the sequence grabber component calls this function and sets this value to 0.

### DESCRIPTION

A sequence grabber component calls your `SGPanelSetResFile` function in order to pass you your component's resource file reference number. By default, the sequence grabber component opens your component's resource file for you. You can use this reference number to retrieve resources from your resource file.

The sequence grabber component also calls this function when it closes your component's resource file. In this case, it sets the `resRef` parameter to 0. Note that the sequence grabber component may close your resource file at any time; you should not count on any particular calling sequence.

If you do not want the sequence grabber component to open your resource file, set the `channelFlagDontOpenResFile` component flag to 1.

## SGPanelGetDITL

---

The `SGPanelGetDITL` function allows a sequence grabber component to determine the dialog items managed by your panel component. The sequence grabber uses this information to build the sequence grabber settings dialog box for the user.

```
pascal ComponentResult SGPanelGetDITL (SeqGrabPanelComponent s,
                                         Handle *ditl);
```

<code>s</code>	Identifies the sequence grabber component's connection to your panel component.
<code>ditl</code>	Contains a pointer to a handle that is to receive your component's item list. Your component should resize this handle as appropriate.



DESCRIPTION

A sequence grabber component calls your `SGPanelGetDITL` function in order to obtain the list of dialog items supported by your panel component. The sequence grabber then places these items into the settings dialog box and presents the dialog box to the user. When the sequence grabber builds the settings dialog box, it places your items appropriately—you do not need to specify particular locations for the items.

Your component returns the item list in a handle that is provided by the sequence grabber component. Note that the sequence grabber component will dispose of this handle after retrieving the item list, so make sure that the item list is not stored in a resource. If your item list is in a resource handle, you can use the Resource Manager's `DetachResource` routine to convert that resource handle into a handle that is suitable for use with the `SGPanelGetDITL` function.

The sequence grabber component will open your resource file before calling this function unless you have instructed the sequence grabber component not to open your resource file (that is, you have set the `channelFlagDontOpenResFile` component flag to 1).

SGPanelInstall

A sequence grabber component calls your `SGPanelInstall` function after adding your items to the settings dialog box, just before it displays the dialog box to the user.

```
pascal ComponentResult SGPanelInstall (SeqGrabPanelComponent s,
                                         SGChannel c, DialogPtr d,
                                         short itemOffset);
```

s

Identifies the sequence grabber component's connection to your panel component.

c

Identifies a connection to the sequence grabber channel associated with your panel component.

d

Contains a dialog pointer identifying the settings dialog box. Your component may use this value to manage its part of the dialog box.

itemOffset

Specifies the offset to your panel's first item in the dialog box. Because sequence grabber components build your dialog items into a larger dialog box containing other items, this value may be different each time your panel component is installed; do not rely on it being the same.

DESCRIPTION

A sequence grabber component calls your `SGPanelInstall` function just before displaying the dialog box to the user. The sequence grabber provides you with information identifying the channel that your panel is to configure, the dialog box, and the offset of your panel's items into the dialog box. You may use this opportunity to set default dialog values or to initialize your control values.

**SEE ALSO**

Sequence grabber components call your component's `SGPanelRemove` function before they remove your panel from the settings dialog box. That function is discussed next.

**SGPanelRemove**

---

Sequence grabber components call your component's `SGPanelRemove` function before removing your panel from the settings dialog box.

```
pascal ComponentResult SGPanelRemove (SeqGrabPanelComponent s,
                                       SGChannel c, DialogPtr d,
                                       short itemOffset);
```

- |                         |   |
|-------------------------|---|
| <code>s</code>          | Identifies the sequence grabber component's connection to your panel component.               |
| <code>c</code>          | Identifies a connection to the sequence grabber channel associated with your panel component. |
| <code>d</code>          | Contains a dialog pointer identifying the settings dialog box.                                |
| <code>itemOffset</code> | Specifies the offset to your panel's first item in the dialog box.                            |

**DESCRIPTION**

A sequence grabber component calls your `SGPanelRemove` function just before removing your items from the settings dialog box. The sequence grabber provides you with information identifying the channel your panel is to configure, the dialog box, and the offset of your panel's items into the dialog box. You may use this opportunity to save any changes you may have made to the dialog box or to retrieve the contents of `TextEdit` items.

If the sequence grabber opened your resource file, it will still be open when it calls this function.

**SEE ALSO**

Sequence grabbers call your `SGPanelInstall` function (described in the previous section) before displaying the settings dialog box to the user.

## Processing Your Panel's Events

When your panel component is loaded into the settings dialog box and active, you may receive and process dialog events and mouse clicks.

Your component's `SGPanelEvent` function acts like a modal-dialog filter function, allowing you to process individual dialog events. The sequence grabber calls your `SGPanelItem` function whenever the user clicks a dialog item.

Whenever the user clicks the OK button, the sequence grabber calls your `SGPanelValidateInput` function. Your panel component may then validate the user's settings.

## SGPanelItem

Your `SGPanelItem` function allows your component to receive and process mouse clicks in the settings dialog box.

```
pascal ComponentResult SGPanelItem (SeqGrabPanelComponent s,  
                                     SGChannel c, DialogPtr d,  
                                     short itemOffset,  
                                     short itemNum);
```

s	Identifies the sequence grabber component's connection to your panel component.
c	Identifies a connection to the sequence grabber channel associated with your panel component.
d	Contains a dialog pointer identifying the settings dialog box.
itemOffset	Specifies the offset to your panel's first item in the dialog box.
itemNum	Contains the item number of the dialog item selected by the user. Note that this is an absolute item number; the sequence grabber does not adjust this value to account for the offset to your first dialog item.

### DESCRIPTION

A sequence grabber component calls your `SGPanelItem` function whenever the user clicks an item in the settings dialog box. Your component may then perform whatever processing is appropriate, depending upon the item number. Note that the sequence grabber provides an absolute item number. It is your responsibility to adjust this value to account for the offset to your panel's first item in the dialog box.

**SEE ALSO**

Your component can filter all dialog events with your `SGPanelEvent` function. This function is described next.

Sequence grabber components use your component's `SGPanelValidateInput` function to validate the current input settings as a whole. That function is discussed on page 7-23.

**SGPanelEvent**

---

Your `SGPanelEvent` function allows your component to receive and process dialog events. This function is similar to a modal-dialog filter function.

```
pascal ComponentResult SGPanelEvent (SeqGrabPanelComponent s,
                                     SGChannel c, DialogPtr d,
                                     short itemOffset,
                                     EventRecord *theEvent,
                                     short *itemHit,
                                     Boolean *handled);
```

<code>s</code>	Identifies the sequence grabber component's connection to your panel component.
<code>c</code>	Identifies a connection to the sequence grabber channel associated with your panel component.
<code>d</code>	Contains a dialog pointer identifying the settings dialog box.
<code>itemOffset</code>	Specifies the offset to your panel's first item in the dialog box.
<code>theEvent</code>	Contains a pointer to an event structure. This event structure contains information identifying the nature of the event.
<code>itemHit</code>	Contains a pointer to a field that is to receive the item number in cases where your component handles the event. The number returned is an absolute, not a relative number, so it must be offset by the <code>itemOffset</code> parameter.
<code>handled</code>	Contains a pointer to a Boolean value. Set this Boolean value to indicate whether your component handles the event: set it to <code>true</code> if you handle the event; set it to <code>false</code> if you do not.

**DESCRIPTION**

A sequence grabber component calls your `SGPanelEvent` function whenever an event occurs in the settings dialog box. Your `SGPanelEvent` function is similar to a modal-dialog filter function. The main difference is that, rather than returning a Boolean value to indicate whether you handled the event, your `SGPanelEvent` function sets a Boolean

value that is provided by the calling function. If you handle the event, be sure to update the field referred to by the `itemHit` parameter.

SEE ALSO

Your component can process mouse clicks with your `SGPanelItem` function. This function is discussed on page 7-21.

SGPanelValidateInput

Sequence grabber components call your component's `SGPanelValidateInput` function in order to allow you to validate the contents of the user dialog box.

```
pascal ComponentResult SGPanelValidateInput
                                (SeqGrabPanelComponent s,
                                Boolean *ok);
```

- `s` Identifies the sequence grabber component's connection to your panel component.
- `ok` Contains a pointer to a Boolean value. You set this Boolean value to indicate whether the user's settings are acceptable. Set it to `true` if the settings are OK; otherwise, set it to `false`.

DESCRIPTION

A sequence grabber component calls your `SGPanelValidateInput` function in order to allow you to validate the settings chosen by the user. This is your opportunity to validate the settings in their entirety, including those for which you may not have received dialog events or mouse clicks. For example, if your panel component uses a TextEdit box, you should validate its contents at this time. Be sure to give the user some indication of what to do to fix the settings.

The sequence grabber calls this function when the user clicks the OK button. If the user clicks the Cancel button, the sequence grabber does not call this function.

You indicate whether the settings are acceptable by setting the Boolean value referred to by the `ok` parameter. If you set this Boolean value to `false`, the sequence grabber component ignores the OK button in the dialog box.

SEE ALSO

Your component can process mouse clicks with your `SGPanelItem` function, described on page 7-21. Your component can filter all dialog events with your `SGPanelEvent` function, described in the previous section.

## Managing Your Panel's Settings

---

Sequence grabber components store their configuration information in Movie Toolbox user data items (see the chapter “Movie Toolbox” in *Inside Macintosh: QuickTime* for more information about user data items). This configuration information includes settings for each of the channels used by the sequence grabber. Because your panel component configures sequence grabber channels, your panel component is responsible for creating and formatting the contents of its user data items. The sequence grabber component calls your component whenever it wants to retrieve these settings. The sequence grabber may also use previously stored settings to restore your panel's settings. This section discusses the functions that allow the sequence grabber to work with your panel's settings.

The sequence grabber calls your `SGPanelGetSettings` function in order to retrieve your panel's current settings. The sequence grabber uses your `SGPanelSetSettings` function to restore those settings to some previous values.

## SGPanelGetSettings

---

Sequence grabber components call your component's `SGPanelGetSettings` function in order to retrieve your panel's current settings.

```
pascal ComponentResult SGPanelGetSettings
                                (SeqGrabPanelComponent s,
                                 SGChannel c, UserData *ud,
                                 long flags);
```

s	Identifies the sequence grabber component's connection to your panel component.
c	Identifies a connection to the sequence grabber channel associated with your panel component.
ud	Contains a pointer to a user data item. Your component is responsible for creating a new user data item and returning that item by means of this pointer. Your component is not responsible for disposing of the user data item.
flags	Reserved for future use.

### DESCRIPTION

A sequence grabber component calls your `SGPanelGetSettings` function in order to obtain a copy of your panel's current settings. The sequence grabber stores these settings for you and may use them to restore your panel's settings by calling your `SGPanelSetSettings` function (described next). Your component should store

whatever values are necessary to properly configure your associated channel component. For example, Apple's video compression panel component saves such values as video compressor component type, compression quality, key frame rate, and frame rate values.

These settings may be stored as part of a larger sequence grabber configuration and may be stored for a long period of time. Therefore, you should not store values that may change without your knowledge (such as component ID or connection values).

You are free to format the data in the user data item in any way you desire. Make sure you can retrieve the settings information from the user data item when your `SGPanelGetSettings` function is called. You may choose to format the data in such a way that other components can parse it easily, thus allowing your component to operate with other panel components.

You create a new user data item by calling the Movie Toolbox's `NewUserData` function (see the chapter "Movie Toolbox" in *Inside Macintosh: QuickTime* for more information about this function). You may then use other Movie Toolbox functions to manipulate the user data item.

SEE ALSO

Sequence grabber components use your component's `SGPanelSetSettings` function to restore this configuration information. That function is discussed next.

SGPanelSetSettings

Sequence grabber components call your component's `SGPanelSetSettings` function in order to restore your panel's current settings.

```
pascal ComponentResult SGPanelSetSettings
                                (SeqGrabPanelComponent s,
                                 SGChannel c, UserData ud,
                                 long flags);
```

s	Identifies the sequence grabber component's connection to your panel component.
c	Identifies a connection to the sequence grabber channel associated with your panel component.
ud	Identifies a user data item that contains new settings information for your panel. Your component must not dispose of this user data item.
flags	Reserved for future use.

## Sequence Grabber Panel Components

**DESCRIPTION**

A sequence grabber component calls your `SGPanelSetSettings` function in order to restore your panel's settings. The sequence grabber may call this function when the user cancels the settings dialog box.

Your component originally creates the settings information when the sequence grabber calls your `SGPanelGetSettings` function (described in the previous section). The sequence grabber passes this configuration information back to you in the `ud` parameter to this function. Your component should parse the configuration information and use it to establish your panel's current settings.

Note that your component may not be able to accommodate the original settings. For example, because the settings may have been stored for some time, the hardware environment may not be able to support the values in the settings. You should try to make your new settings match the original settings as closely as possible. If you cannot get close enough, return an appropriate sequence grabber or sequence grabber channel result code.

You may use Movie Toolbox functions to manipulate the user data item (see the chapter "Movie Toolbox" in *Inside Macintosh: QuickTime* for more information about functions that work with user data items).

**RESULT CODES**

`noDeviceForChannel`     -9408     Device cannot support settings

Other appropriate sequence grabber or sequence grabber channel result codes

**SEE ALSO**

Sequence grabber components use your component's `SGPanelGetSettings` function (described in the previous section) to retrieve the configuration information.



## Summary of Sequence Grabber Panel Components

---

### C Summary

---

#### Constants

---

```

/* component type value */
#define SeqGrabPanelType 'sgpn'    /* panel component type */

/* component manufacturer code values */
#define SeqGrabCompressionPanelType 'sour' /* input source selection */
#define SeqGrabSourcePanelType 'cmpr' /* compression settings */

/* componentFlags values for sequence grabber panel components */
enum {
    channelFlagDontOpenResFile = 2, /* do not open resource file */
    channelFlagHasDependency    = 4 /* needs special hardware */
};

enum {
    /* sequence grabber panel request codes */
    kSGCPanelGetDitlSelect      = 0x200, /* SGPanelGetDITL */
    kSGCPanelCanRunSelect       = 0x202, /* SGPanelCanRun */
    kSGCPanelInstallSelect      = 0x203, /* SGPanelInstall */
    kSGCPanelEventSelect        = 0x204, /* SGPanelEvent */
    kSGCPanelItemSelect         = 0x205, /* SGPanelItem */
    kSGCPanelRemoveSelect       = 0x206, /* SGPanelRemove */
    kSGCPanelSetGrabberSelect    = 0x207, /* SGPanelSetGrabber */
    kSGCPanelSetResFileSelect    = 0x208, /* SGPanelSetResFile */
    kSGCPanelGetSettingsSelect   = 0x209, /* SGPanelGetSettings */
    kSGCPanelSetSettingsSelect   = 0x20A, /* SGPanelSetSettings */
    kSGCPanelValidateInputSelect = 0x20B /* SGPanelValidateInput */
};

```

## Functions

---

### Managing Your Panel Component

```
pascal ComponentResult SGPanelSetGrabber
    (SeqGrabPanelComponent s, SeqGrabComponent sg);

pascal ComponentResult SGPanelCanRun
    (SeqGrabPanelComponent s, SGChannel c);

pascal ComponentResult SGPanelSetResFile
    (SeqGrabPanelComponent s, short resRef);

pascal ComponentResult SGPanelGetDITL
    (SeqGrabPanelComponent s, Handle *ditl);

pascal ComponentResult SGPanelInstall
    (SeqGrabPanelComponent s, SGChannel c,
     DialogPtr d, short itemOffset);

pascal ComponentResult SGPanelRemove
    (SeqGrabPanelComponent s, SGChannel c,
     DialogPtr d, short itemOffset);
```

### Processing Your Panel's Events

```
pascal ComponentResult SGPanelItem
    (SeqGrabPanelComponent s, SGChannel c,
     DialogPtr d, short itemOffset, short itemNum);

pascal ComponentResult SGPanelEvent
    (SeqGrabPanelComponent s, SGChannel c,
     DialogPtr d, short itemOffset,
     EventRecord *theEvent, short *itemHit,
     Boolean *handled);

pascal ComponentResult SGPanelValidateInput
    (SeqGrabPanelComponent s, Boolean *ok);
```

### Managing Your Panel's Settings

```
pascal ComponentResult SGPanelGetSettings
    (SeqGrabPanelComponent s, SGChannel c,
     UserData *ud, long flags);

pascal ComponentResult SGPanelSetSettings
    (SeqGrabPanelComponent s, SGChannel c,
     UserData ud, long flags);
```

## Pascal Summary

---

### Constants

---

```

CONST
    {component type value}
    SeqGrabPanelType           = 'sgpn';    {panel component type}
    {component manufacturer code values}
    SeqGrabCompressionPanelType = 'comp';   {compression settings}
    SeqGrabSourcePanelType      = 'sour';   {input source selection}

    {componentFlags values for sequence grabber panel components}
    channelFlagDontOpenResFile  = 2;       {do not open resource file}
    channelFlagHasDependency    = 4;       {channel has special hardware}

    {sequence grabber panel component request codes}
    kSGCPanelGetDitlSelect      = $200;    {SGCPanelGetDitl}
    kSGCPanelCanRunSelect       = $202;    {SGCPanelCanRun}
    kSGCPanelInstallSelect      = $203;    {SGCPanelInstall}
    kSGCPanelEventSelect        = $204;    {SGCPanelEvent}
    kSGCPanelItemSelect         = $205;    {SGCPanelItem}
    kSGCPanelRemoveSelect       = $206;    {SGCPanelRemove}
    kSGCPanelSetGrabberSelect   = $207;    {SGCPanelSetGrabber}
    kSGCPanelSetResFileSelect   = $208;    {SGCPanelSetResFile}
    kSGCPanelGetSettingsSelect  = $209;    {SGCPanelGetSettings}
    kSGCPanelSetSettingsSelect  = $20A;    {SGCPanelSetSettings}
    kSGCPanelValidateInputSelect = $20B;    {SGCPanelValidateInput}

```

### Routines

---

#### Managing Your Panel Component

```

FUNCTION SGPanelSetGrabber (s: SeqGrabComponent; sg: SeqGrabComponent):
    ComponentResult;

FUNCTION SGPanelCanRun      (s: SeqGrabComponent; c: SGChannel):
    ComponentResult;

FUNCTION SGPanelSetResFile  (s: SeqGrabComponent; resRef: Integer):
    ComponentResult;

FUNCTION SGPanelGetDITL    (s: SeqGrabComponent; VAR ditl: Handle):
    ComponentResult;

```

## Sequence Grabber Panel Components

```

FUNCTION SGPanelInstall      (s: SeqGrabComponent; c: SGChannel;
                             d: DialogPtr; itemOffset: Integer):
                             ComponentResult;

FUNCTION SGPanelRemove      (s: SeqGrabComponent; c: SGChannel;
                             d: DialogPtr; itemOffset: Integer):
                             ComponentResult;

```

**Processing Your Panel's Events**

```

FUNCTION SGPanelItem        (s: SeqGrabComponent; c: SGChannel;
                             d: DialogPtr; itemOffset: Integer;
                             itemNum: Integer): ComponentResult;

FUNCTION SGPanelEvent       (s: SeqGrabComponent; c: SGChannel;
                             d: DialogPtr; itemOffset: Integer;
                             VAR theEvent: EventRecord;
                             VAR itemHit: Integer;
                             VAR handled: Boolean): ComponentResult;

FUNCTION SGPanelValidateInput
                             (s: SeqGrabComponent; VAR ok: Boolean):
                             ComponentResult;

```

**Managing Your Panel's Settings**

```

FUNCTION SGPanelGetSettings (s: SeqGrabComponent; c: SGChannel;
                             VAR ud: UserData; flags: LongInt):
                             ComponentResult;

FUNCTION SGPanelSetSettings (s: SeqGrabComponent; c: SGChannel;
                             ud: UserData; flags: LongInt): ComponentResult;

```

**Result Codes**


---

noDeviceForChannel	-9408	Cannot work with specified channel
badComponentSelector	0x80008002	Function not supported