

# QuickTime™ Conferencing SDK

## QTC Configuration Files

---

## Contents:

About Configuration Files.....	2
QuickTime Conferencing Configuration Files .....	2
Service, Transport and Network Types.....	2
Configuration Resources .....	4
An Example—Adding a Service Type.....	4
An Example—Adding a Network Component .....	5
Document Version.....	7
Credits .....	7
Important Note.....	7

# About Configuration Files

---

QuickTime Conferencing applications and components may require special configuration information in order to work. You are required to create a set of configuration resources if any of the following are true:

- If you are an application developer and wish to create a new service type, other than the two built in service types (VideoPhone and Broadcaster), and wish to work with the PowerTalk business cards for QuickTime Conferencing. Your configuration resource will enable your new service to appear as a pop-up menu item inside business cards.
- If you are a transport /network component developer and wish to work with the Apple Videophone application and/or the PowerTalk template for QuickTime Conferencing. This configuration information is used by the Apple Videophone application when it calls the Conference/Browser component API (specifically in the MTConferenceListen and MTBrowserBrowse calls.)

An application developer creating a new service type, may choose not to create any configuration information, and instead pass strings to MTConferenceListen and MTBrowserBrowse as done by SeeWorld applications. In this case

- a. Your application will be able to use non-PowerTalk browsers,
- b. Your application will be able to browse online with PowerTalk but will be unable to add or create QuickTime Conferencing addresses of your service type in the PowerTalk Personal Catalog.

This document describes the format of the configuration resources contained in the QuickTime Conferencing system extension and QuickTime Conferencing Configuration files.

## QuickTime Conferencing Configuration Files

---

QuickTime Conferencing Configuration files reside in the Preferences folder within the System Folder of a user's machine. The Configuration files are of type: 'mtcf'. Developers can optionally set the creator type to the default type of 'mtlk', which causes the Finder to display the Configuration file with the standard QuickTime Conferencing Configuration file icon. Developers are free to use their own file creator types and display their own icons if appropriate.

## Service, Transport and Network Types

---

QuickTime Conferencing Configuration files can contain one or more descriptions for new network, transport and service types. These descriptions are stored in resources within the configuration files.

Currently, there are four resource types for describing configuration information within QuickTime Conferencing Configuration files:

Resource Type	Format	Configuration Information
'svcs'	Pascal string	User Readable Service Name
'nets'	Pascal string	User Readable Network Name
'trns'	Pascal string	User Readable Transport Name
'brcg'	configuration string	Service Configuration String

Each service type provided by the developer should have a corresponding four byte unique identifier often described as an OSType. Typically, developers can use the same OSType for their service type as the creator type for their application.

A corresponding localized string exists in human readable form described by the following resource.

```
type 'svcs' {
    String;      /* human readable service name string */
};
```

An example service name string resource might be the following:

```
resource 'svcs' (128, "tnhl") {
    "Town Hall" /* user readable name */
};
```

In this case, the service OSType is 'tnhl' and the name of the service would be "Town Hall".

The above resource has to be a named resource and the name of the resource must be of the format {Service OSType}. (For example, for service type 'brod' the resource name will be "brod".) The contents of the resource is the localized user-readable name (Pascal string) of the service type represented by the service OSType. e. g 'Broadcaster'.

Similarly, creators of new transports and networks should create 'trns' and 'nets' resources to provide user readable strings for configuration purposes.

```
type 'trns' {
    String;      /* human readable service name string */
};

type 'nets' {
    String;      /* human readable service name string */
};

resource 'trns' (128, "mors") {
    "Morse Code" /* user readable transport name */
};

resource 'nets' (128, "smke") {
    "Smoke Signals" /* user readable network name */
};
```

## Configuration Resources

---

Developers will also need to express how a network component should be configured to use a particular service with a transport. This triplet of Service-Transport-Network will have a configuration string associated with it. Triplets that do not have configuration strings defined are assumed to be invalid, and the user should not be able to specify them. (For example, the PowerTalk template for QuickTime Conferencing will only allow valid combinations of Service-Transport-Network to be created within a Media Conferencing PowerTalk card. For each triplet of {Service OSType}{Transport Sub-type}{Network sub-type} there exists a resource that describes the network specific configuration. Examples of configuration strings might be a TCP port number or an NBP registration type string.

Configuration strings are expressed as configuration resources of resource type 'brcg'. Each configuration resource contains the transport type for that configuration, the network type of that configuration and a variable length string containing the configuration information. The 'brcg' resource is described as follows.

```
type 'brcg' {
    OSType;      /* transport sub-type */
    OSType;      /* network sub-type */
    char tab = "\0x09";
    String;      /* service type information */
    char newline = "\0x0D";
    char nulchar = "\0x00";
};
```

The 'brcg' resource has to be a named resource- the name must be of the format {Service OSType}. {transport sub-type }{network sub-type}. For example, for a configuration where the transport type is 'mtlk', the network type is 'isdn' and service type 'brod' the resource name will be 'brodmtlkisdn'.

Using our "Town Hall" example again, a service string might be created for "Town Hall" services using the MovieTalk transport component and AppleTalk network component as follows:

```
resource 'brcg' ( 128, "thnlmtlkatlk") {
    'mtlk',      /* transport type */
    'atlk',      /* network type */
    /* implicit tab character */
    "TownHall"  /* AppleTalk NBP name */
    /* implicit newline and null character */
};
```

## An Example—Adding a Service Type

---

As an example, let's create an imaginary new service for our "Town Hall" application that will use QuickTime Conferencing. Following are the steps to add this new service type for QuickTime Conferencing to find it.

1. Define a service name. Let us call it "TownHall".
2. Define a corresponding service OSType. Let us call it 'tnhl'.

3. Create a configuration file (that should be installed in the preference folder) of type 'mtcf' and creator 'mtlk'. Let us call this file TownHallQTCPrefs.

4. Add the following resources to TownHallQTCPrefs.

```
resource 'brcg' ( 128, "thnlmtlkatlk") {
    'mtlk',      /* transport type */
    'atlk',      /* network type */
                /* implicit tab character */
    "TownHall"  /* AppleTalk NBP name */
                /* implicit new line and null character */
};

resource 'svcs' (128, "tnhl") {
    "Town Hall" /* user readable name */
};
```

If the TownHall service is supported on other network and transport sub-types more 'brcg' resources are needed. For each transport, network sub-type pair a new named resource of type 'brcg' needs to be created.

The PowerTalk Browser reads the service type string and uses it for validating the service type embedded in the user selected address. See "Browser Components" for more information.

If you have created your resources properly, the TownHall service type will appear in the service type pop-up menu inside the QuickTime Conferencing business card

## An Example—Adding a Network Component

When you add new network and transport components you need to create similar resources. As mentioned above, each service type is represented by a unique OSType. A corresponding localized string exists in human readable form. For each triplet of {Service OSType}{Transport Sub-type}{network sub-type} there exists a resource that describes the network specific configuration. If you are adding a transport component 'mytr' you will need to create resources with names that are a concatenation of {Service OSType}'mytr' {network sub-type}. If you are adding a network component 'mynw' you will need to create resources with names that are a concatenation of {Service OSType}{Transport sub-type}'mynw'.

As an example if you are creating a network component here are the steps you need to follow. Let us call the network component sub-type 'mynw'.

1. Identify all existing services and transport pairs you intend to support.
2. For each service/transport you plan to support create the service type string (e.g. a port number or a service name).
3. Create resources with names being concatenation of {service OSType}{Transport sub-type}'mynw'.

An example of a resource name for service type 'tnhl' and transport sub-type 'mtlk' would be 'tnhlmtnlkmywn'. Let us assume that the service port is 189. Notice that the resource has it in ASCII form.

```
resource 'brcg' ( 128, "tnhlmtnlkmywn") {
    'mtlk',      /* transport type */
    'mywn',      /* network type */
    /* implicit tab character */
    "189"        /* Network info */
                /* implicit new line and null character */
};
```

Here is the actual hex data resource:

```
data 'brcg' (128, "tnhlmtnlkmywn") {
    $"6D74 6C6B 6D79 6E77 0931 3839 0D00"
    /* mtnlkmywn 189\r. */
};
```

## Document Version

---

Version 1.0

## Credits

---

Document by Dean Blackketter and Murali Ranganthan

## Important Note

---

The content of this document is of a preliminary nature, and is subject to change without notice.

This document and the software described in it are copyrighted, with all rights reserved. Under the copyright laws, this document or the software may not be copied, in whole or part, without written consent of Apple, except in the normal use of the software or to make a backup copy of the software. The same proprietary and copyright notices must be affixed to any permitted copies as were affixed to the original. Under the law, copying includes translating into another language or format.

The Apple logo, QuickTime, PowerTalk, and QuickTime Conferencing are registered trademarks of Apple Computer, Inc.