

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

» Estás en: Inicio » Tutoriales » Integración de MonkeyTalk en iOS



Ignacio Acisclo Pérez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)

Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2014-11-12

Tutorial visitado 1 veces [Descargar en PDF](#)

Integración de MonkeyTalk en iOS

Para los que no sepan que es monkeyTalk, se trata de una herramienta para automatizar flujos de navegación en nuestras apps, es decir, nos sirve para testear la interfaz de usuario.

Para empezar lo primero que tenemos que hacer es crearnos una cuenta en [CloudMonkey](#) y después descargarnos el IDE desde [este enlace](#).


Como este tutorial no trata sobre desarrollar en iOS voy a pedirlos que os descarguéis la app de ejemplo sobre la que vamos a probar MonkeyTalk y en la que ya están incluidas la librerías que hay añadir al proyecto así que solo será necesario incluirlas en caso de sigáis el tutorial con una app propia.

Descarga de app ejemplo aqui:
<https://github.com/iAcisclo/MonkeyTalkDemo>

Bien una vez descargado el zip desde la web tenemos dos cosas importante la carpeta /agents y la carpeta /MonkeyTalkIDE. En la carpeta agents vamos a tener la librería que tenemos que incluir a nuestro target de pruebas y en la carpeta MonkeyTalkIDE tenemos el entorno que nos brinda CloudMonkey para crear nuestros proyectos de pruebas y scripts.

Lo primero que vamos hacer es duplicar el target de nuestra app que queremos probar, que va a ser el tarjeta donde incluiremos las siguientes librerías:

- libstdc++.6.0.9.dylib (del sistema)
- libsqlite3.dylib (del sistema)
- ICFNetwork.framework (del sistema)
- QuartzCore.framework (del sistema)
- libMonkeyTalk-2.0.9.a (se encuentra en la carpeta /agentes antes mencionada)

▼ Linked Frameworks and Libraries		
Name	Status	
 libMonkeyTalk-2.0.9.a	Required	⬇
 QuartzCore.framework	Required	⬇
 CFNetwork.framework	Required	⬇
 libsqlite3.dylib	Required	⬇
 libstdc++.6.0.9.dylib	Required	⬇
+ -		

Una vez incluidas las librerías necesarias vamos a editar nuestro Scheme para renombrar el target:



Síguenos a través de:



Últimas Noticias

» [Curso JBoss de Red Hat](#)

» [Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris](#)

» [Portales, gestores de contenidos documentales y desarrollos a medida](#)

» [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)

» [Screencasts de programación narrados en Español](#)

[Histórico de noticias](#)

Últimos Tutoriales

» [Soporte de Redis con Spring: RedisTemplate](#)

» [Embeber vídeo en MailChimp](#)

» [Tutorial VIPER en Swift](#)

» [Monitorización de Apache](#)

Kafka

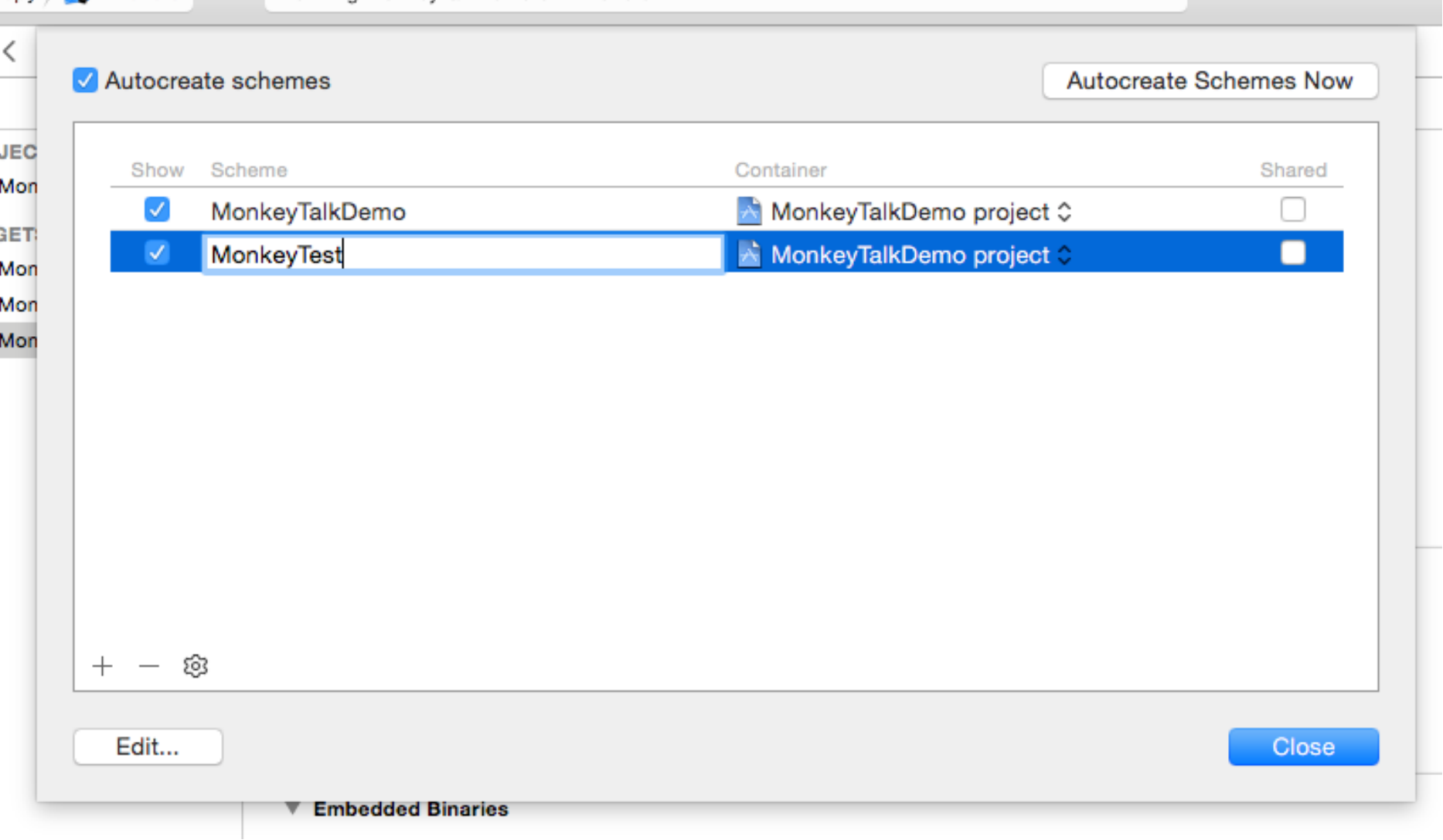
» [Hooks en Cordova: Cargar todos los plugins de forma automática](#)

Últimos Tutoriales del Autor

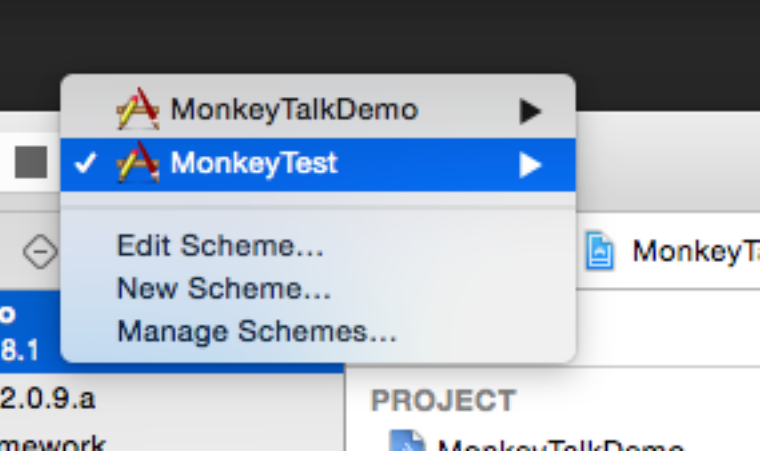
» [Tutorial VIPER en Swift](#)

» [Transiciones personalizadas en iOS7](#)

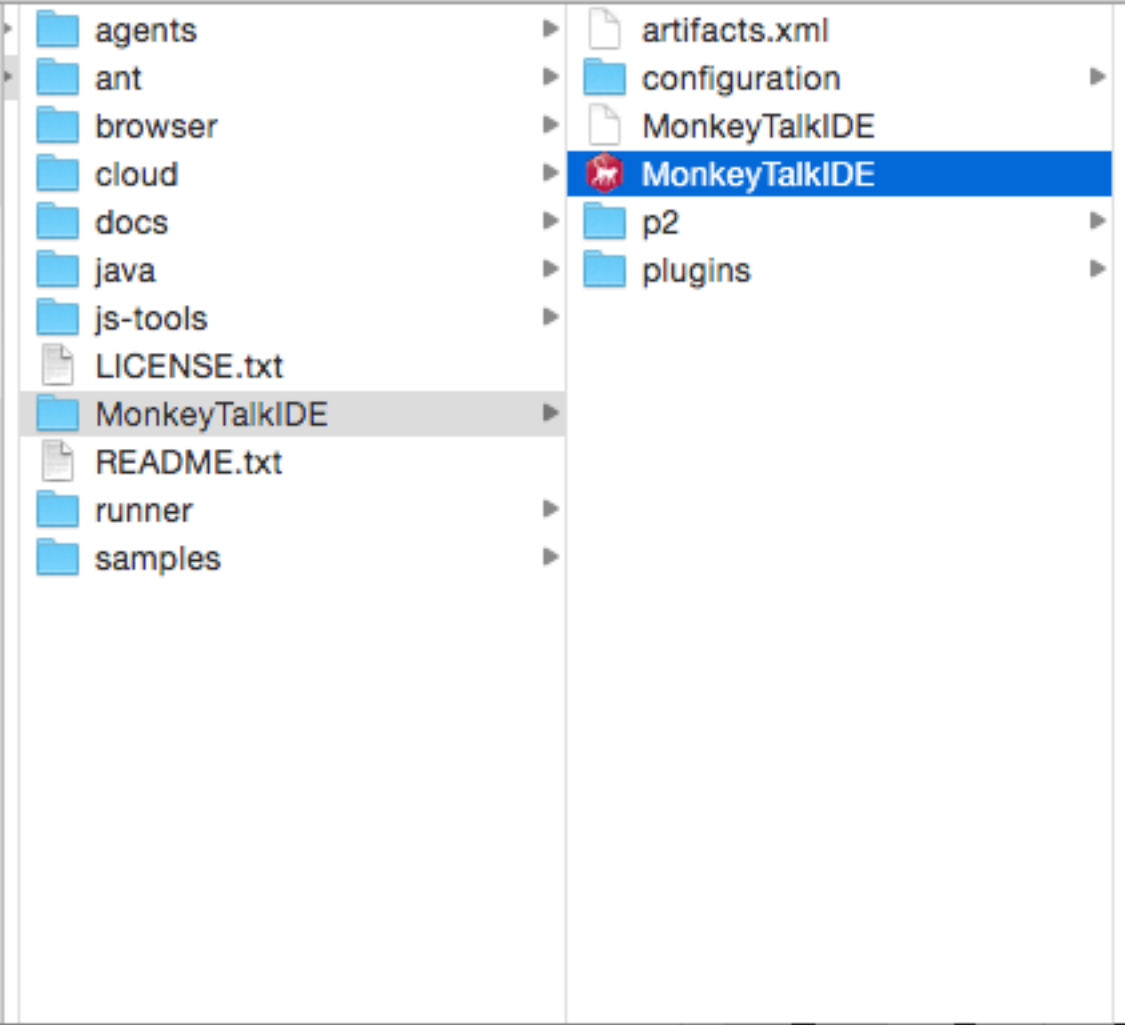
» [Notificaciones locales en iOS.](#)



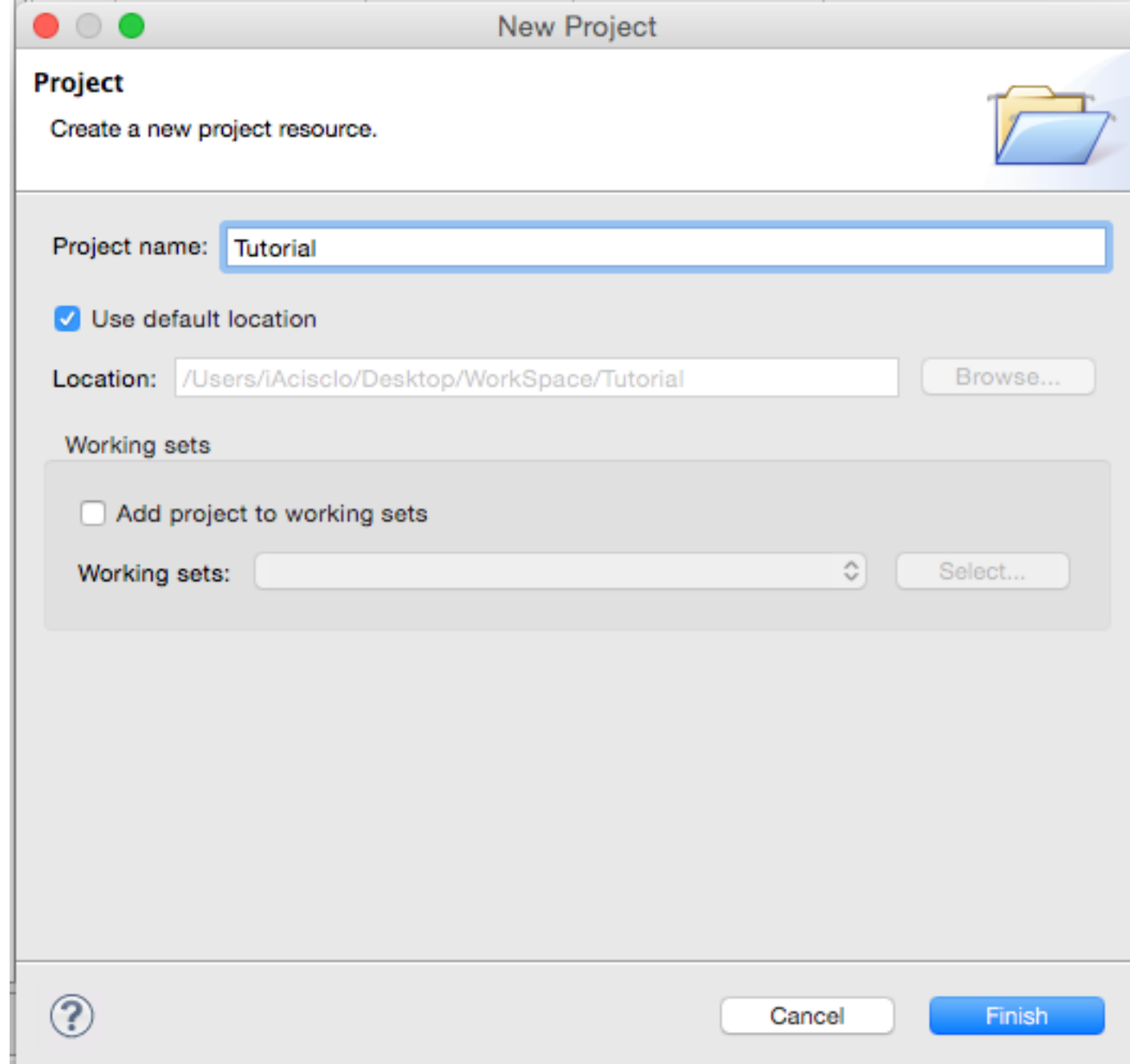
y lo seleccionamos como target a ejecutar en la compilación:



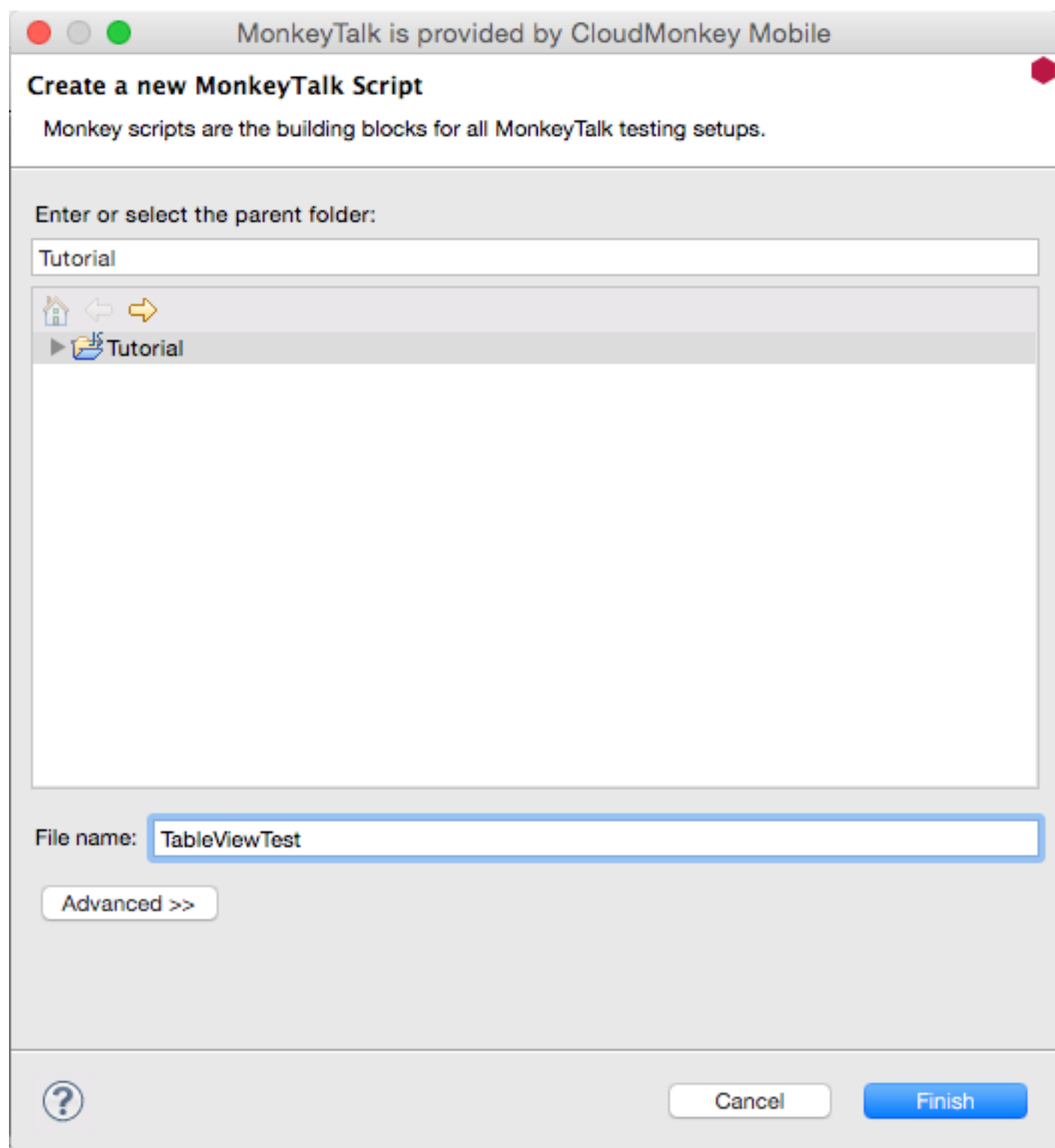
Ahora vamos a abrir el IDE de monkeyTalk



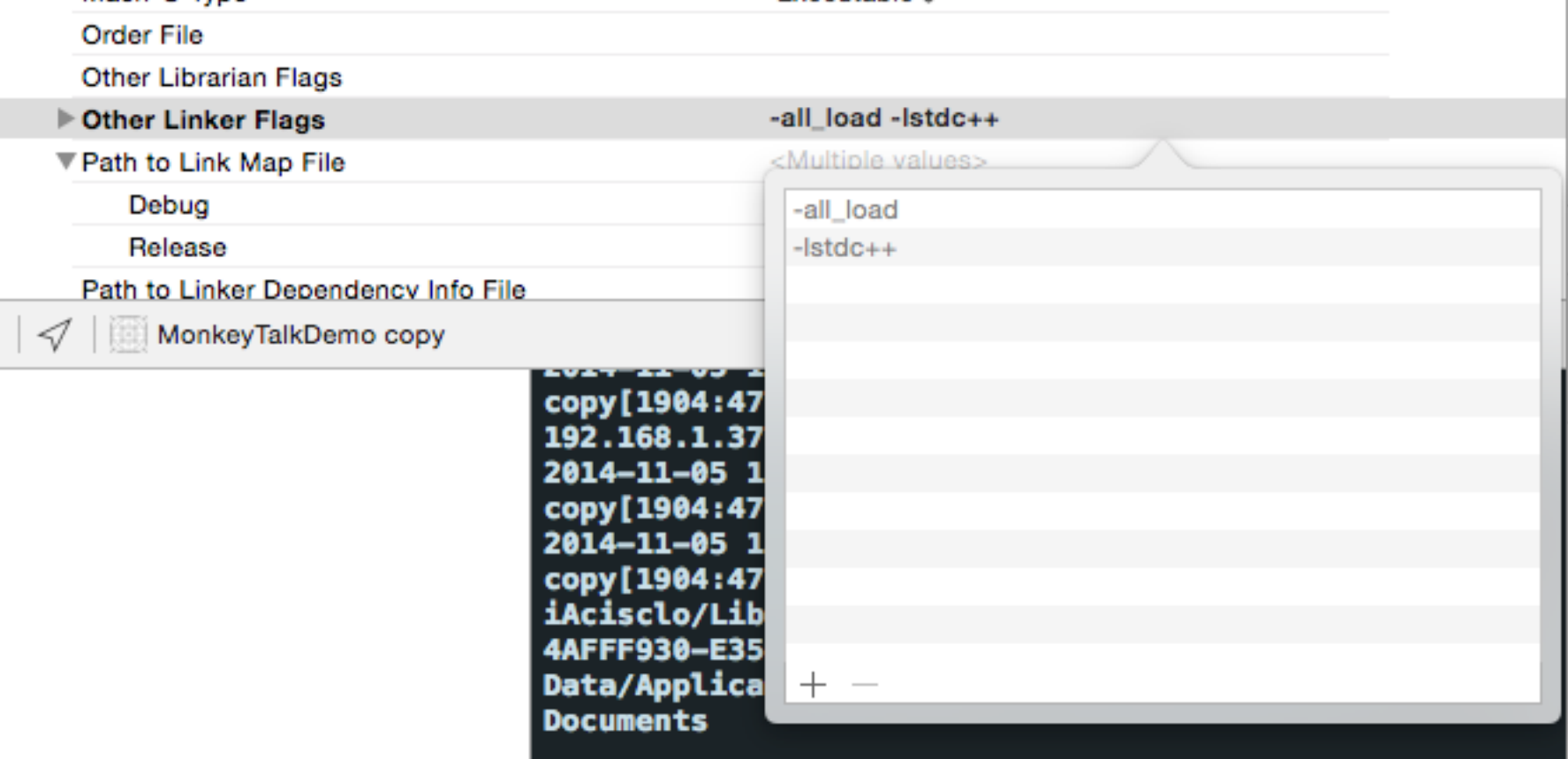
Un vez abierto vamos a File/New/MonkeyTalk Project y creamos un nuevo proyecto:



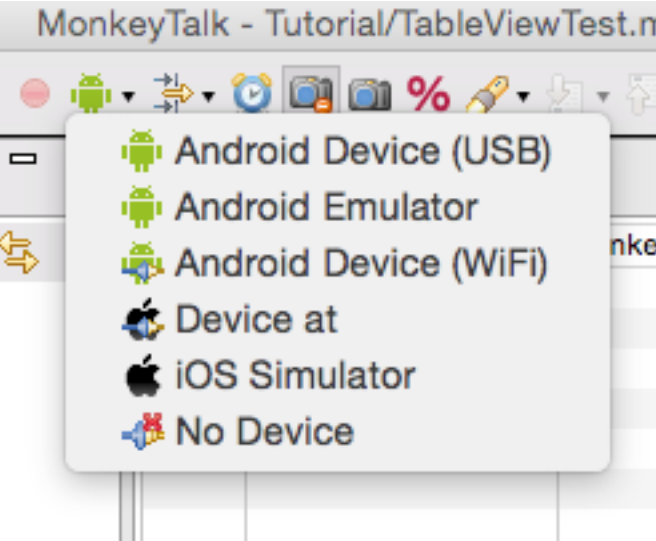
Vamos a necesitar generar dos Scripts, uno para definir nuestros movimientos por la pantalla y otro para iniciar la simulación, así que vamos a File/New/Script y creamos un script que vamos a llamar TableViewTest que va a ser el que registre los movimientos en nuestra tabla:



Una vez hecho esto vamos a nuestro proyecto de Xcode y nos vamos a buildSettings de nuestro Project y buscamos en la sección de Linking la fila “other linker flags” y añadimos -lstdc++ y _all_load:



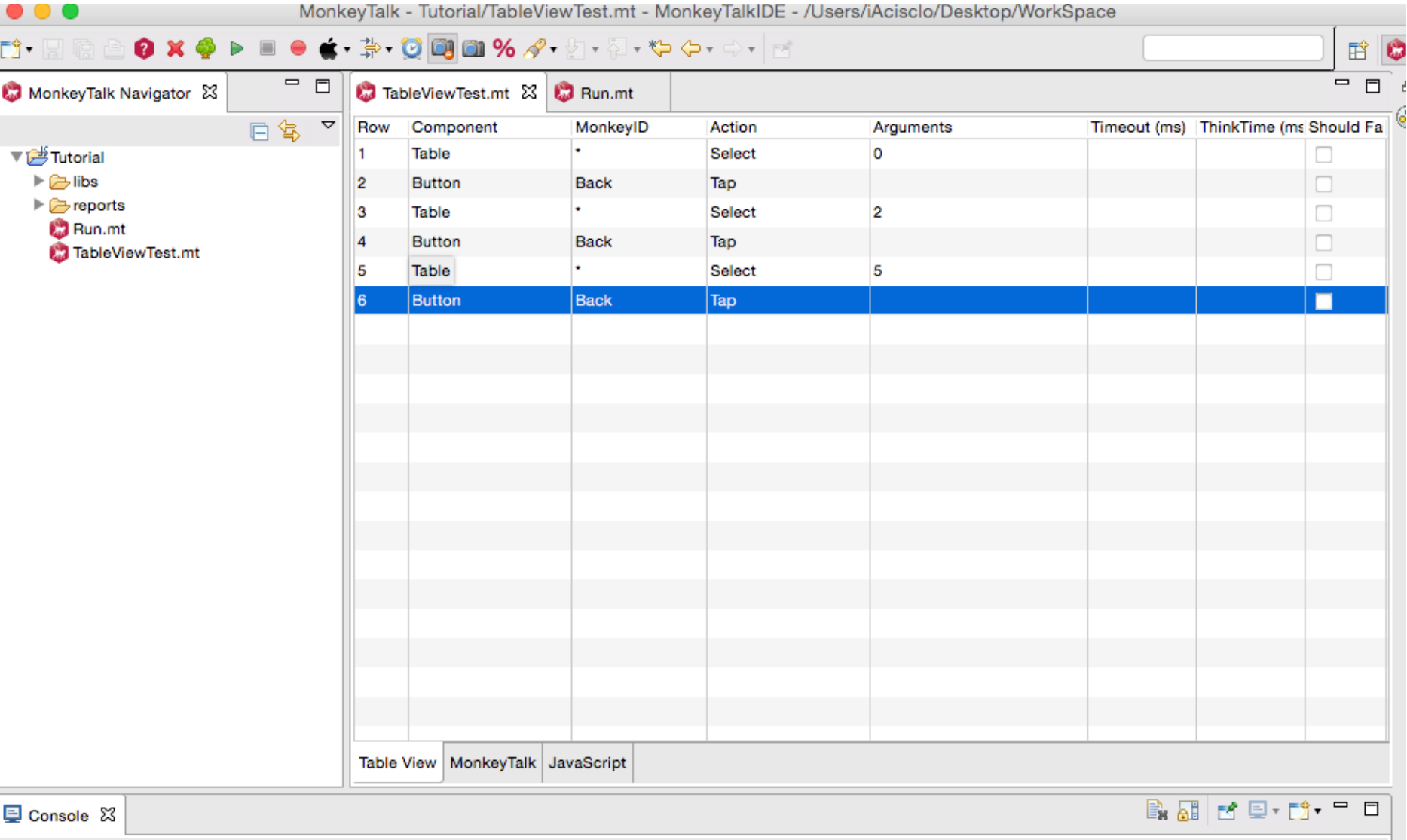
y pulsamos command-R para arrancarla en el simulador, cuando esté arrancada vamos al MonkeyTalk IDE y en un botón en la parte superior desplegamos las opciones de linkado seleccionando iOS Simulator:



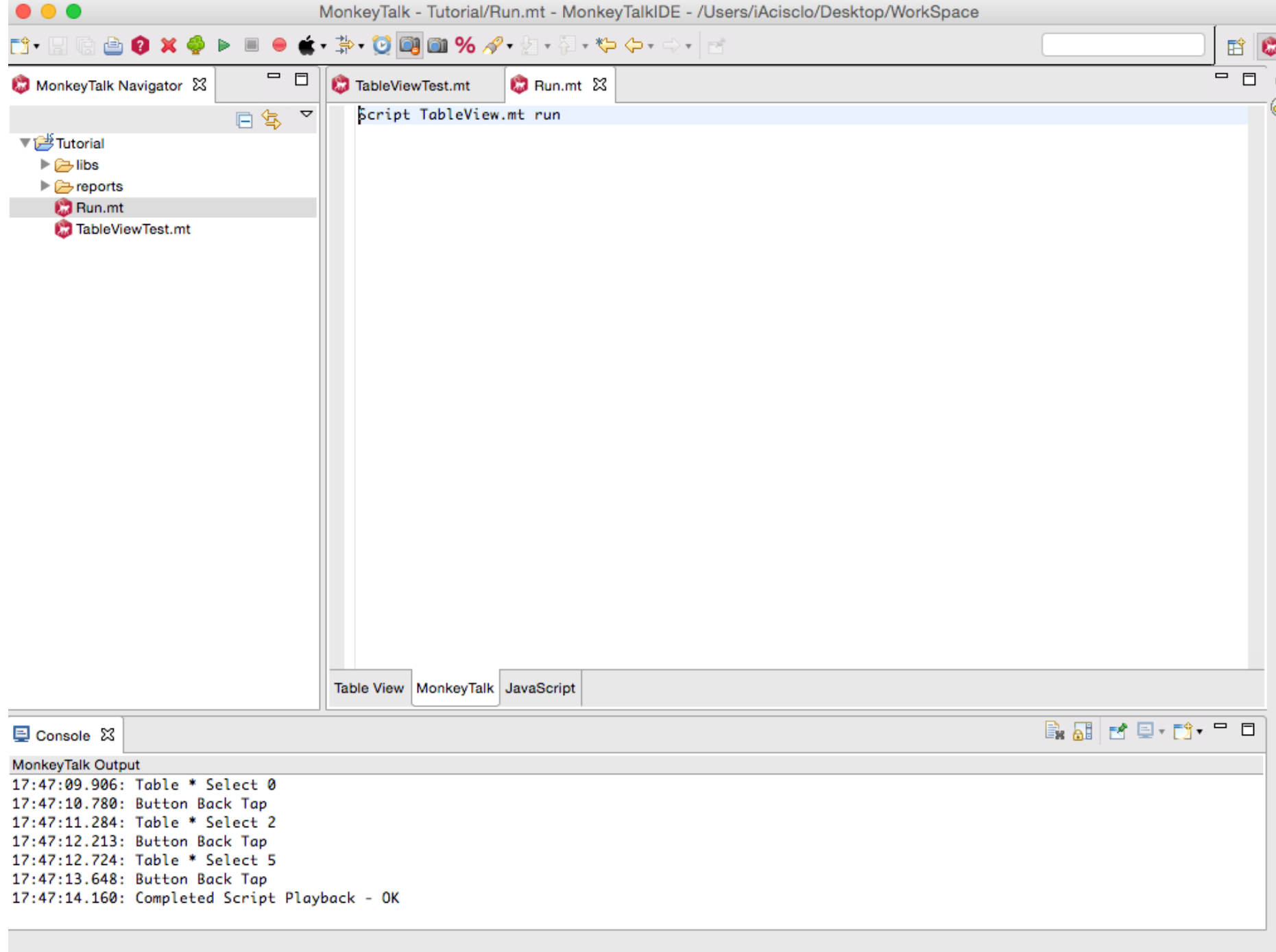
En las trazas de la consola del IDE deberíamos ver “ Connection type set to: iOS Simulator” que nos indica que ya está linkado correctamente.

Bien vamos a grabar los movimientos que queremos testar así que en la barra superior del MonkeyTalk IDE tenemos un botón con un círculo redondo que indica “record”, lo pulsamos y nos vamos al simulador y hacemos los movimientos que queramos que se registren y cuando acabemos pulsamos en el botón cuadrado negro de Stop.

Como podréis comprobar en el Script TableViewTest se han registrado una serie de entradas que son los clics que hemos realizado en nuestra app:



Ahora vamos a crear un nuevo Script que llamaremos “Run” y nos vamos a su pestaña “MonkeyTalk” que se encuentra en la parte inferior del IDE y escribimos lo siguiente: Script TableViewTest.mt run



Y ya esta! ahora solo hay que pulsar sobre el botón play y el solito ejecutara los mismo movimientos que hemos registrado antes.

Como podéis ver es muy sencillo de configurar y nos ofrece la posibilidad de automatizar flujos completos de navegación de nuestra app.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)



Por favor, vota +1 o compártelo si te pareció interesante

Share | 0 0

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

PUSH THIS

Page Pushers

[Community](#)

[Help?](#)

no clicks

0 people brought clicks to this page



powered by [karmacracy](#)

