

«interface»
contract::ISocketConnectorListener
+ connectionEstablished(Socket) : void
+ error(int, Object) : void

«interface»
contract::ISocketConnector
+ cancel() : void
+ finishedListening() : boolean
+ getConnectionDetails() : ConnectionParameter
+ listen() : void



<i>AsyncTask</i>
impl::SocketConnector
- dc: IHostDependencyContainer {readOnly}
- LOG_TAG: String = "SocketConnector" {readOnly}
- port: volatile int
+ cancel() : void
+ doInBackground(Object) : AsyncTaskResult<Socket>
+ finishedListening() : boolean
+ getConnectionDetails() : ConnectionParameter
+ getLocalIpAddress() : String
+ listen() : void
+ onPostExecute(AsyncTaskResult<Socket>) : void
+ onPreExecute() : void
+ SocketConnector(IHostDependencyContainer)