

# Контроль качества канала публикации/проигрывания

В сборке [2.6.110](#) добавлена возможность получения от сервера сообщений с текущим битрейтом публикации и проигрывания и [контроля качества канала](#), основываясь на разнице битрейтов клиента и сервера.

Чтобы включить возможность контроля качества канала, необходимо использовать метод `WCSStream.enableConnectionQualityCalculation()`

```
publishStream?.enableConnectionQualityCalculation(true);
```

Затем необходимо зарегистрировать callback функцию, которая получает на вход показатель качества и текущие значения битрейта клиента и сервера, усредненные фильтром Калмана

```
publishStream?.onConnectionQualityCallback({currentQuality, clientFiltered, serverFiltered in  
    self.updateQualityStatus(currentQuality, view: self.publishQuality);  
});
```

Пример использования:

регистрация обработчика [code](#), обработчик [code](#)

```
@IBOutlet weak var publishQuality: UILabel!  
...  
publishStream?.enableConnectionQualityCalculation(true);  
publishStream?.onConnectionQualityCallback({currentQuality, clientFiltered, serverFiltered in  
    self.updateQualityStatus(currentQuality, view: self.publishQuality);  
});  
...  
fileprivate func updateQualityStatus(_ quality:kFPWCSConnectionQuality, view: UILabel) {  
    switch (quality) {  
        case .fpwcsConnectionQualityBad:  
            view.text = "BAD";  
            view.textColor = .red;  
            break;  
        case .fpwcsConnectionQualityGood:  
            view.text = "GOOD";  
            view.textColor = .yellow;  
            break;  
        case .fpwcsConnectionQualityPerfect:  
            view.text = "PERFECT";  
            view.textColor = .green;  
            break;  
        case .fpwcsConnectionQualityUnknown:  
            view.text = "UNKNOWN";  
            view.textColor = .darkText;  
    }  
}
```