|  |  |
| --- | --- |
| **Title：** | **SOA Gateway MRD** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Document Owner** | **Changes** |
| <0.1> | 2020-02-27 | 丁云荣 | 初稿 |
| <0.2> | 2020-03-03 | 丁云荣 | 对consumer/provider的消息处理逻辑进行分组 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1 需求介绍 3

2 需求内容 4

2.1 车型配置及差异 4

2.2 需求概述 4

2.3 需求详述 5

2.3.1 SOAGateway集成 5

2.3.2 SOAGateway Client开发 6

2.3.2.1 Consumer开发 7

2.3.2.2 Provider开发 9

2.3.2.3 Protobuf message 10

# 需求介绍

本文档用于整理Ford SOA 集成/实现相关需求，参考福特提供如下文档:

* fnv\_\_soa\_ (a) General Information.pdf
* fnv\_\_soa\_ (b)API Frequently Asked Questions.pdf
* fnv\_\_soa\_ (c) Writing a SOA Service Consumer (from ECG).pdf
* fnv\_\_soa\_ (d) Writing SOAServiceDirectoryManager.pdf
* fnv\_\_soa\_ (e) Writing a SOA Service Provider (from ECG).pdf
* fnv\_\_soa\_ (f) Writing a SOA on-demand data broadcast Consumer or Provide.pdf
* fnv\_\_soa\_ (g) Gateway (from ECG).pdf
* fnv\_\_soa\_ (h) Run-time Version Checking for API Messages.pdf
* Service Manager System Service Descriptor (from ECG).pdf
* IVI-SoA Network Guidelines v1.1 May 4, 2018.pdf
* FNV2 In Vehicle SOA Middleware (FNV2-SOA) (v1.0).pdf
* FNV2-SOA Gateway (v0.2).pdf
* FNV2-SOA Gateway Detailed Design.pdf
* FNV2-SOA Security Architecture (v0.3).pdf
* SOA Gateway Deployment and User Guide.pdf
* SoaGateway\_desay.pptx

# 需求内容

## 车型配置及差异

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | CD542 | | CX727 | U725 | P702 | U554 | U625ICA |
| 功能 | Ambient/Trend | Titanium/ST Line | Mid/High | Ford Bronco SUV | Ford Raptor F-150  Pickup truck | Lincoln Navigator  Large SUV (3 row seat) | Explorer |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

所有车型均包含此功能，可视为无功能差异

## 需求概述

SOA - Service Oriented Architecture, 中文可以翻译为面向服务架构, 其主要目的是基于车身以太网搭建一个基于订阅/发布机制的系统，用来在各个ECU(包括ECG/SYNC/TCU等)之间进行通信. 从下面的架构图可以看到，SYNC为soa的client端，ECG为soa的server(broker)端。其中在SYNC内部又分两部分：SOAGateway，主要用来跟ECG通信；

SOAGateway Client主要通过内部的IPC机制跟SOAGateway进行交互。





在此功能下SYNC端主要需要做两方面的事情：

1. 集成Ford发布的SOAGateway可执行程序(包括依赖库、配置文件等)，确保其对外可以跟ECG，对内可以跟SOAGateway client正常通信。
2. 实现SOAGateway的client端，以便各个基于SOA架构的feature可以通过SOAGateway client跟SOAGateway进行通信，从而连到ECG的broker上。

## 需求详述

### SOAGateway集成

福特释放的soa package主要包含两部分内容：

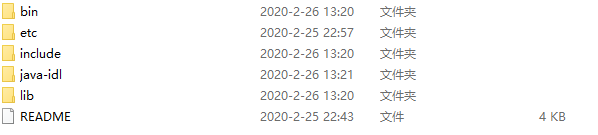
1. SOAGateway

这里面主要包括SOAGateway的可执行程序，其依赖库，以及配置文件等内容，德赛需要将其预置到系统中，并且保证其可以正常开机自动启动运行

1. 开发SOAGateway client所需的动态库以及相关头文件集成

部分feature福特会直接释放SOAGateway Client端的动态库，部分feature需要德赛基于相应feature的SPSS要求以及福特提供的api 头文件和soa/proto库进行相应的consumer/provider的开发，SOAGateway Client相关的库等文件也需要集成到系统中供最终APP调用。目前明确Ford会提供库的feature为ippt/fci/ccs/wir，其余需要德赛基于soagateway进行开发，由于各feature开发不归属与SOAGateway中，此处不再单独列出。

福特释放版本的顶级目录结构如下：



### SOAGateway Client开发

SOA client端的开发主要分为两部分工作：

一种是Ford提供了具体feature api和相应的so库，比如libippt/libwir.so这种形式的，德赛基于这些API封装接口给APP使用，此处暂不讨论这类feature的。

另一种是feature的SPSS里有SOA消息描述，但是福特没有提供相应api的so库的，需要德赛基于soa的consumer和provider机制进行api封装，以便app可以调用相关接口跟ECG通信，比如SVS/CE-DTE等，此文档主要描述此类feature的基础consumer和provider的开发方式。

Client开发中会有相应log输出以便调试。

下图是德赛基于整体SOAGateway的理解设计的软件架构图：

DsvNDKWrapper层会对各个feature以及soa的基本api进行封装，保证功能齐全的前提下尽可能接口简单，保证后面fnv2Halservice到fnv2service到fnv2manager的开发工作相对简单。



#### Consumer开发

Consumer在soa架构中主要有如下功能：

1、建立连接，依托于SoaMessageManager同SOAGateway建立连接

2、消息订阅，用来接收服务端(provider)发布出来的消息，包括subscribe/unsubscribe,接收消息更新

3、远程调用(remoteCall)，主要用于consumer端主动向provider端发起请求时使用，包括发送请求消息和接收响应消息。

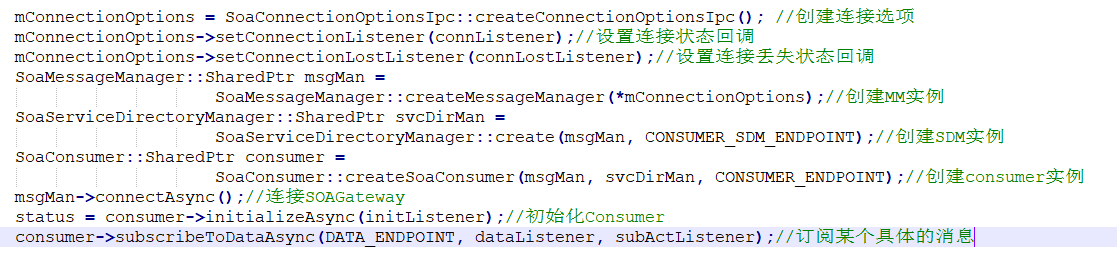
4、订阅/反订阅服务状态

Consumer的开发主要包括几部分：

1. consumer的初始化

consumer的典型初始化流程如下：

初始化主要分为同SOAGateway(及ECG的broker)连接初始化和进行具体的消息订阅，主要代码见下面代码片段

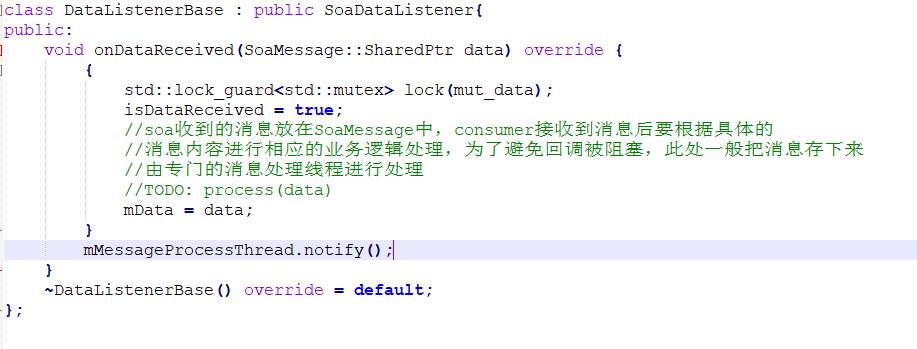


1. 消息处理

Consumer开发中主要关注的消息处理有两种:一种是订阅类消息的处理；一种是主动请求消息的处理

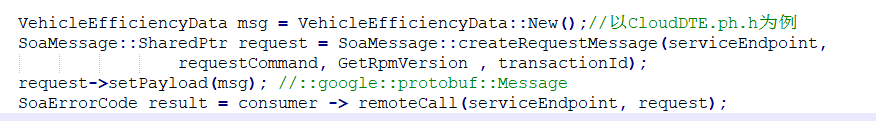
* 1. 接收订阅消息并处理

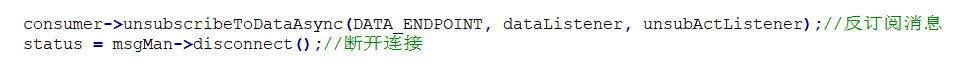
如上初始化时consumer订阅了DATA\_ENDPOINT指向的service所描述的消息，当provider端进行新消息发布的时候，dataListener就会收到provider端的消息回调



* 1. 主动调用provider的消息接口

Consumer主动给Provider发送消息主要通过remoteCall接口，remoteCall类似正常的函数调用，可以发送请求消息，也可以同步/异步获取到对端的响应消息，此接口有两种形式，同步和异步remoteCallAsync和remoteCall，典型操作如下：

 3．Consumer退出及异常处理

Consumer释放流程主要包括反订阅消息、断开MessageManager连接，以及其他资源释放等 

Consumer使用过程中可能存在于SOAGateway连接断开的场景，通过SoaMessageManager的回调，以及相关调用的报错信息可以知道具体的错误信息，consumer的实现需要能够正确处理这些异常，确保可以恢复，进行正确错误提示等。

#### Provider开发

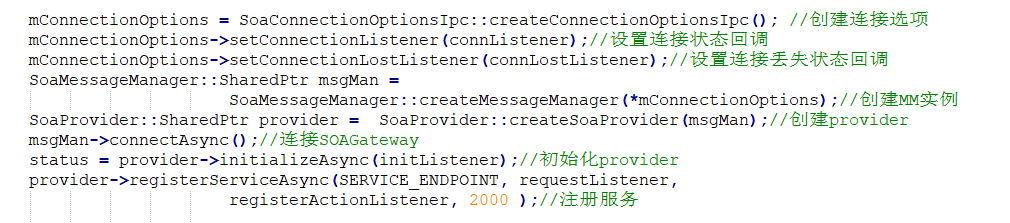
Provider在soa架构中主要有三个作用：

1. 消息发布，用来向已经订阅消息的consumer发布消息
2. Service注册，向broker注册自己的服务，以便consumer可以主动发起请求
3. 处理consumer的消息请求，并返回相应的结果

按照Provider的生命周期执行顺序，开发主要包括几部分：

1. Provider的初始化

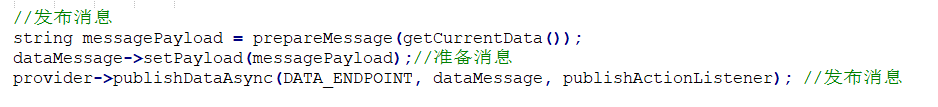
初始化过程主要包含初始化连接和服务的注册



1. 消息处理

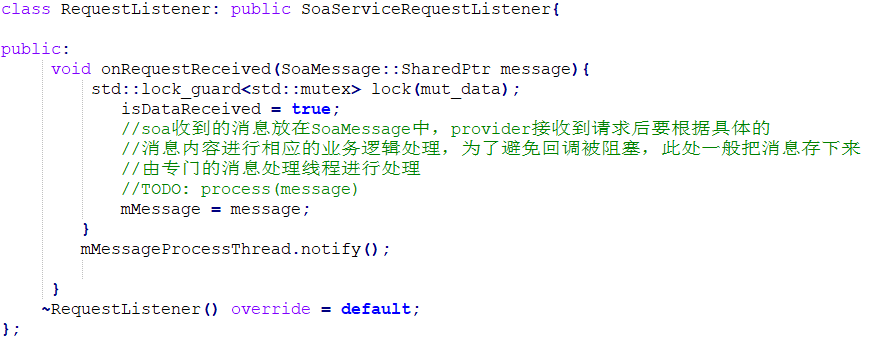
Provider的消息处理主要包括：主动发布消息和接收consumer的消息请求并正确响应

* 1. 主动发布消息

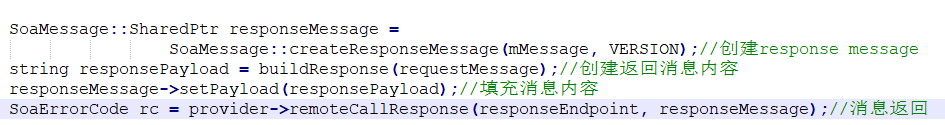


* 1. 接收consumer的消息请求并处理

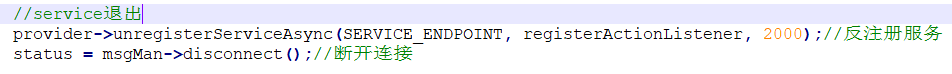
接收：



响应：



1. Provider退出以及异常处理
   1. Provide释放流程主要包括反注册服务、断开MessageManager连接，以及其他资源释放等



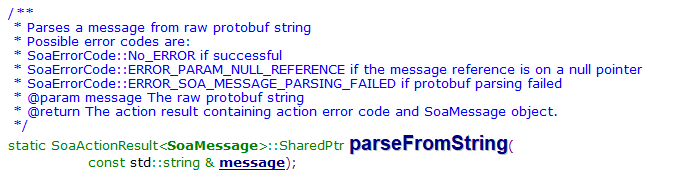
* 1. Provider在运行中也可能存在跟SOAGateway断开等各种异常情况，provider开发中要能正确处理这些异常，具体参考program guide里的crash management章节。

#### Protobuf message

Protobuf是Google定义的跨系统的数据传递机制，我们主要使用它的消息的序列化和反序列化机制，

SoaMessage跟protobuf::Message的关联主要采用下面几种方式

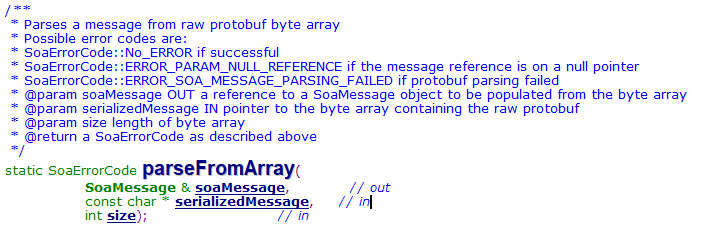
1. parseFromString



message由::google::protobuf::Message:: SerializeToString生成



1. parseFromArray



message由::google::protobuf::Message:: SerializeToArray生成



1. SoaMessage直接关联::google::protobuf::Message

