

fnv::soa: (d) Writing SOAServiceDirectoryManager (from ECG)

- [DEPRECATED](#)
- [Summary](#)
- [Registering System Services](#)
 - [The API](#)
 - [The API Usage](#)

DEPRECATED

Summary

[SOAConsumer](#) APIs rely on service contract and service status data cached in `SOAServiceDirectoryManager` instance. `SOAConsumer` needs to know a service status and an endpoint to start talking to a service. `SOAConsumer` internally uses `SOAServiceDirectoryManager` API to get the service status and the endpoint. Upon a request, `SOAServiceDirectoryManager` contacts `SOAServiceDirectory` service to get the latest service state updates and caches them. We suggest to create a single instance of `SOAServiceDirectoryManager` and share it between all the consumers in the component to avoid extra memory usage and SOA traffic.

Registering System Services

If `SOAServiceDirectoryManager` gets a service status request and there is no cached data found it performs a remote call to Service Directory service. This happens silently for Dynamic services. However, Service Directory service do not track system services which are always running to avoid unnecessary extra SOA traffic. Therefore, there is no way for `SOAServiceDirectoryManager` to automatically resolve system services status. We provide a special API to allow service consumer developers to populate `SOAServiceDirectoryManager` with well-knowns system service metadata.

The API

```
/**
 * Use this API to statically register system services you are going to access in you code.
 * System service assumption:
 *     It is a well knows service with the contract defined in a header file
 *     The service has global lifecycle
 * @param contract The contract
 * @return The error code
 */
SOAErrorCode SOAServiceDirectoryManager::registerSystemService(SOAServiceContract::SharedPtr contract,
SOAServiceStatus status)
```

The API Usage

We assume that developers put `SOAServiceContract` definitin for test provider in a public header file.

API Usage

```
// This should be defined in an external header file
const SOAServiceContract::SharedPtr testServiceContract(new SOAServiceContract(SOAClientEndpoint::
createClientEndpointMqtt("TEST_TOPIC")));

// Creating a service directory manager
SOAMessageManager::SharedPtr messageManager;
SOAClientEndpoint::SharedPtr sdEndpoint; SOAServiceDirectoryManager::SharedPtr sdManager =
SOAServiceDirectoryManager::create(sdEndpoint, messageManager);

// Registering system test service.
sdManager->registerSystemService(testServiceContract, SOAServiceStatus::RUNNING);
```