std::ofstream in\_test;

Json::Value value\_test;

Json::Value value\_read;

Json::Reader reader\_test;

value\_test["msgBody"] = "guanhe";

value\_test["msgHead"] = "head";

in\_test.open("server\_rtsp\_connect\_req.json", std::ios\_base::out);

if (!in\_test.is\_open()) {

return -1;

}

in\_test << value\_test.toStyledString() << endl;

in\_test.close();

std::ifstream in\_w;

in\_w.open("server\_rtsp\_connect\_req.json", std::ios::binary);

if (!in\_w.is\_open()) {

return -1;

}

//if (reader\_test.parse(in\_w, &value\_read)) {

//

//}

if (reader\_test.parse(in\_w, value\_read)) {

printf("[%s] [%d] msgBody === %s\n", \_\_FUNCTION\_\_, \_\_LINE\_\_, (char \*)value\_read["msgBody"].toStyledString().data());

}

//判断文件是否存在

std::ifstream in\_w;

in\_w.open("server\_rtsp\_connect\_req.json", std::ios::in);

if (!in\_w.is\_open()) {

}