

1.rdp_client_reset_codecs设置编码类型对象, 目前看有三种分别是rfx h.264 nsc

```
static BOOL rdp_client_reset_codecs(rdpContext* context)
{
    rdpSettings* settings;

    if (!context || !context->settings)
        return FALSE;

    settings = context->settings;
    context->codecs = codecs_new(context);

    if (!context->codecs)
        return FALSE;

    return freerdp_client_codecs_prepare(context->codecs, FREERDP_CODEC_ALL,
                                         settings->DesktopWidth, settings->DesktopHeight);
}
```

- codecs_new创建codecs对象, 仅仅是创建一块内存

```
settings = context->settings;
context->codecs = codecs_new(context);
```

- freerdp_client_codecs_prepare主要是创建了codecs成员的内存分配, 以及成员初值的设置里面逻辑很复杂, 目前不是很清楚。

```
return freerdp_client_codecs_prepare(context->codecs, FREERDP_CODEC_ALL,
                                     settings->DesktopWidth, settings->DesktopHeight);
```

2.目前不清楚是做什么的

```
winpr_InitializeSSL(flags);
```

3.设置nego部分成员, 包括hostname port protocol status等

```
nego_init(rdp->nego);
nego_set_target(rdp->nego, settings->ServerHostname, settings->ServerPort);
```

4. 根据是否启用gateway功能设置cookie

```
if (settings->GatewayEnabled) { ... }
else
{
    status = nego_set_cookie(rdp->nego, settings->Username);
}
```

5.给nego成员设置初值

```

nego_set_send_preconnection_pdu(rdp->nego, settings->SendPreconnectionPdu);
nego_set_preconnection_id(rdp->nego, settings->PreconnectionId);
nego_set_preconnection_blob(rdp->nego, settings->PreconnectionBlob);
nego_set_negotiation_enabled(rdp->nego, settings->NegotiateSecurityLayer);
nego_set_restricted_admin_mode_required(rdp->nego, settings->RestrictedAdminModeRequired);
nego_set_gateway_enabled(rdp->nego, settings->GatewayEnabled);
nego_set_gateway_bypass_local(rdp->nego, settings->GatewayBypassLocal);
nego_enable_rdp(rdp->nego, settings->RdpSecurity);
nego_enable_tls(rdp->nego, settings->TlsSecurity);
nego_enable_nla(rdp->nego, settings->NlaSecurity);
nego_enable_ext(rdp->nego, settings->ExtSecurity);

```

6. 设置nego最大缓存, 为什么要设置缓存以及大小, 目前还不清楚

```

nego_set_cookie_max_length(rdp->nego, settings->CookieMaxLength);

```

7. 设置nego的routing_token的值, 作用现在还不清楚

```

if (settings->LoadBalanceInfo)
{
    if (!nego_set_routing_token(rdp->nego, settings->LoadBalanceInfo, settings->LoadBalanceInfoLength))
        return FALSE;
}

```

8. 设置rdp的state

```

rdp_client_transition_to_state(rdp, CONNECTION_STATE_NEGO);

```

9. 创建socket建立链接, 设置自身支持和选择的协议。nego发送消息是同步的, 即会阻塞等待response。主要还是选择协议, 是rdp nla tls ext中的某一个。发送自身支持和选择的协议, 然后等待服务器决定协议并接受服务器数据, 最终根据服务器的选择确定协议。这里面也创建与服务器通信的socket通过nego_transport_connect函数再nego_attempt_*系列的函数中。链接和创建socket的过程在其他文档详述

```

if (!nego_connect(rdp->nego))
{
    if (!freerdp_get_last_error(rdp->context))
    {
        freerdp_set_last_error(rdp->context, FREERDP_ERROR_SECURITY_NEGO_CONNECT_FAILED);
        WLog_ERR(TAG, "Error: protocol security negotiation or connection failure");
    }

    return FALSE;
}

```

10. 获取最终决定好的协议, 如果是ssl或者rdp则设置自动登录

```

SelectedProtocol = nego_get_selected_protocol(rdp->nego);

if ((SelectedProtocol & PROTOCOL_SSL) || (SelectedProtocol == PROTOCOL_RDP))
{
    if ((settings->Username != NULL) && ((settings->Password != NULL) ||
        (settings->RedirectionPassword != NULL && settings->RedirectionPassw
            settings->AutoLogonEnabled = TRUE;
}

```

11. 设置接收数据的回调rdp_recv_callback, 设置socket是否阻塞

```

/* everything beyond this point is event-driven and non blocking */
rdp->transport->ReceiveCallback = rdp_recv_callback;
rdp->transport->ReceiveExtra = rdp;
transport_set_blocking_mode(rdp->transport, FALSE);

```

12. 只要不是nla模式, 就创建mcs链接, msc是在rdp_new中创建的综训msc的协议规范发送链接协议数据。并且转换rpd的state

```
if (rdp->state != CONNECTION_STATE_NLA)
{
    if (!mcs_client_begin(rdp->mcs))
        return FALSE;
}
```

13.接受服务器回复的链接回馈数据, 更详细的说明参见专门的文档

```
while (rdp->state != CONNECTION_STATE_ACTIVE)
{
    if (rdp_check_fds(rdp) < 0)
    {
        if (!freerdp_get_last_error(rdp->context))
            freerdp_set_last_error(rdp->context, FREERDP_ERROR_CONNECT_TRANSPORT_FAILED);

        return FALSE;
    }
    SwitchToThread();
}
```