Redis接口错误处理方案

1. BOOL32 GetConfListByDomain(const s8\* pchDomainInfo, const s8\* pchMtE164,

const BOOL32 bCheckPublic, ConfList& lstConf);

终端获取正在召开会议列表

REDIS异常则返回FALSE，数据为空返回TRUE, lstConf为空

失败未做错误处理，发空会议列表

失败应告知终端错误

6. BOOL32 GetConfInfoByE164(const s8\* achDomainInfo, const s8\* achE164,

TConfDetailInfo& tConfDetail);

获取会议详情

当前REDIS操作失败和没数据 都返回失败。 会NACK

REDIS操作失败和没数据都是返回失败。当前虽然一样的效果但是应该要把两种情况区分。

1. BOOL32 GetBookConf(const s8\* pchConfE164, CRedisBookConf& bookConf);

需改脚本，没有数据回空字符串。 执行失败返回FALSE, 执行成功，根据CRedisBookConf

的e164是否为空判断是否存在该预约会议

1. 获取会议详情

应区分REDIS操作失败和没数据，但错误处理相同

1. 终端提前召开预约会议

应区分REDIS操作失败和没数据，返回不同的错误码，当前都是回会议不存在

1. 结会通知，删除预约会议

再次执行数据库存储过程releaseConf

---不管

1. BOOL32 GetConfListByDomain(const s8\* pchDomainMOID, std::list<TRedisConfInfo>& lstConf);

会议重名处理

当前重命名处理一定成功

会议重名处理也有可能失败，需做错误处理

终端修改会议名，若此时重命名失败，则回NACK

22. BOOL32 GetBookConfList(const s8\* pchDomainMOID, std::vector<CRedisBookConf>& confVec)；

A. 会议重名处理

重名处理可能失败，需做错误处理。

B. 获取预约会议列表

获取预约会议列表，失败应告知终端错误

3. BOOL32 GetConfNumByDomain(const s8\* pchDomainMOID,

const EMConfNumType emConfNumType, u32& dwConfNum);

1. 终端获取会议数量， 若失败则打印错误退出，不回终端

失败应告知终端

B. 检测会议数量是否超过限制

当前处理正确，若失败，则返回ERR\_DB\_OPR

1. BOOL32 SetConfData(const TRedisConfInfo& tConfInfo);
2. 创会

当前处理正确

1. 延长会议是否到时，修改会议时间

未做失败处理，需要添加失败处理，REDIS异常就NACK延长请求

1. 终端修改会议名称请求和通知

未做失败处理，修改会议名称请求失败回NACK

若是会馆改名，MCU只是通知CSS，先不管这种问题 \*\*\*\*

10. BOOL32 UpdateConfData(const TRedisConfInfo& tConfInfo);

A. 恢复会议时更新会议状态

当前失败，则此次恢复取消，处理正确

B. MCU会议列表上报的时候调整REDIS会议数据

当前未做失败处理。若失败，结束此次调整

1. BOOL32 ChangeConfState(const s8\* achMoid, const s8\* achE164, u8 byState);
2. MCU断链，修改该MCU拥有会议状态为待恢复

当前未做失败处理，需要在检测会议状态处理函数中检测所有正在召开会议所属

MCU是否在本地缓存中存在，若不存在，需更改会议状态为待恢复

B. 创会/恢复ACK时，修改会议状态为正在进行

当前未做失败处理，若失败通知MCU结束该会议。仍通知会馆创会成功。

C. 恢复NACK，重新设置会议为待恢复状态

可以不处理。检测会议进行处理

D. 检测会议处理函数检测创会或恢复超时，将会议修改为待恢复

可以不处理。后续仍定时处理，直到成功

15. BOOL32 GetConfListByCmuIp(const s8\* pchCmuIp, std::list<TRedisConfInfo>& lstConfE164);

A. MPC断链

当前失败直接退出处理。需要在检测会议状态处理函数中检测所有正在召开会议所属

MCU是否在本地缓存中存在，若不存在，需更改会议状态为待恢复

B. 报会议列表，调整redis上有但MCU上没有的会

退出处理，会定时去请求会议列表

1. BOOL32 GetChargeJson(const s8\* pchConfE164, std::string& chargeJson);

结会时报话单

当前未做错误处理

若失败，应停止结会。并让MCU报会议列表。通过调整会议列表再次进行结会处理，报话单。构造好话单json先不报，等成功删除会议数据之后再报话单

26. BOOL32 DelConf(const s8\* pchDomainMoid, const s8\* pchE164, const s8\* pchMcuIp);

createconfNack的时候

结会时删除REDIS数据，当前失败未做错误处理

应让MCU重报会议列表，通过调整会议列表再次结会，删除REDIS数据

1. BOOL32 GetConfMonitorsInfo(const s8\* pchE164, std::vector<TMonitorAliveInfo>& vecMonitors);

获取监控信息失败，若失败，则停止此次恢复

8.BOOL32 GetConfExInfo(const s8\* pchE164, TRedisConfInfo& tConfInfo);

接口不动，改脚本。脚本执行失败，回FALSE。没数据回TRUE, 空字符串。执行成功，根据tConfInfo的e164判断是否有会议数据。

1. 终端加入会议判断会议是否存在，若存在通知MCU添加该终端

若失败，通知终端入会错误。若没有该会议，则判断是否是预约会议或公共模板。

1. MCU发给会馆的消息，判断会议是否存在

REDIS异常，是否可以忽略这些消息？没有会议忽略这些消息

1. 会馆发给MCU的会议操作消息，判断会议是否存在

会议不存在和REDIS异常 都是忽略这些消息。顶多操作不生效。

1. 调整MCU上报会议和REDIS会议。

若不存在需通知MCU结束该会，若REDIS操作失败，等待会议列表调整定时器调整

1. 调整会议列表时，结掉在该REDIS上的会议归属的MCU的会议。

没有会议信息和操作失败，都通知MCU重新发送会议列表

1. 删除预约会议时，判断会议是否存在，若该会议已经召开，通知MCU结束该会

若没有会议信息是正常的，若REDIS操作失败，是否忽略？V

1. 会馆结会时判断会议是否存在，存在结会成功，不存在，结会失败。

没有会议信息回结会失败，REDIS异常NACK

1. 通过公共模板创会时判断会议是否存在
2. 通过预约会议创会时判断会议是否存在

若REDIS异常，则返回错误。若会议不存在是正常现象

1. 延长会议获取会议时长信息

REDIS异常和没有会议都NACK延长请求

1. 创会

若REDIS异常，则NACK创会请求，若不存在会议是正常现象

1. MCU修改会议名称时获取会议信息

若REDIS异常，则NACK修改名称请求，若没有会议则通知MCU结会。

1. MCU通知会议名称发生改变时，获取会议信息

暂不处理

1. 创会ACK
2. 创会NACK
3. 恢复NACK
4. 结会通知
5. 构造停止会议监控json

若REDIS异常则取消此次恢复

1. 结会时，构造话单

若REDIS操作失败，构造话单失败，则结会失败

1. 判断会议是否归属MCU

结会通知，创会ACK,NACK，恢复ACK,NACK做对应处理

MCU操作消息：

若REDIS异常，若REDIS正常且不归属则忽略该消息

1. 构造恢复JSON时，获取会议信息

若此时REDIS异常，则取消此次恢复

5. BOOL32 SetMaxDomainConfNum(const s8\* pchDomainMOID,

const u32 dwMaxSmallConfNum, const u32 dwMaxLargeConfNum);

1. AP数据库读取用户域会议数量限制并更新到REDIS上

当前未做失败处理。若失败，则需要利用现有机制让AP数据库重新加载数据，再更新到REDIS上 --打印区分

B. Bmc创建/修改用户域数据

当前未做失败处理，若失败，利用现有机制重新加载ap数据库用户域表数据

--加定时器 LOAD

可优化删除：

1. BOOL32 RecoverTimeOpr(const s8\* pchConfE164, time\_t\* time = NULL);

没问题的接口：

1. BOOL32 IsConfDataExist(const s8\* pchConfE164, BOOL32& bExist);

MCU断链，判断会议是否含有基础数据

若REDIS异常，由检测会议函数进行异常处理

1. BOOL32 IsConfExist(const s8\* abyE164, BOOL32& bExists);

23. BOOL32 GetAllBookConf(std::vector<CRedisBookConf>& confVec);

启动上报UPU。若失败已做错误处理。不断定时去上报。直到成功。

计算预约会议数量上报给NM 填0

1. BOOL32 UpdateMonitorAliveTime(const s8\* pchE164, const s8\* pchIp, const s8\* pchPort);

此次监控心跳无法写入

27. BOOL32 StartInitRedisConf();

失败就因为redis异常。程序启动时候必须成功。才能继续正常业务。失败做定时处理

24.BOOL32 GetCreateConfJson(const std::string& confE164, const BOOL32 bHasXmpu, std::string& createConfJson);

如果redis异常，则返回失败，本次恢复终止。会议状态是1.

如果redis上没有会议数据，则返回空的创会json,则不恢复，因为没有基础数据且状态是1，后面会被删除

16. BOOL32 ConstructE164(const s8\* pchPrefix, const s8\* pchBegin, const s8\* pchEnd,

std::string& strE164);

Redis没连上，构造失败。创会失败

1. BOOL32 GetExpiredMonitors(std::vector<TMonitorAliveInfo>& vecMonitors);

后续redis正常，将删除过期监控

1. BOOL32 GetAllConfExInfo(std::list<TRedisConfInfo>& lstConfInfo);

恢复会议和检测会议时使用。若失败，则此轮恢复和调整取消没问题。

若连接不正常做失败处理

当前虽未做错误处理，但逻辑上没啥问题

1. BOOL32 GetCmuConfNum(const s8\* pchCmuIp, u32& dwNum);

用于分配MCU函数

若失败，则该MCU分配失败，没有问题

19. BOOL32 AddBookConf(const CRedisBookConf& cConfInfo);

20. BOOL32 ModBookConf(const CRedisBookConf& cConfInfo);

21. BOOL32 DelBookConf(const s8\* pchConfE164);

当前未做错误处理，会馆也没有错误处理相关函数

应回NACK，会馆也需做相关错误处理\*\*\*\*\*\*\*\*\*