### ITSM &G2A 合并项目设计文档(Chat Server部分)

##### Customize the transcript email subject

* + 1. 整体描述

Chat Server的改动集中在发邮件的处理逻辑:

聊天结束时

1. 要根据portal端的配置决定是否需要自动给指定邮箱发邮件
2. 邮件的subject 也要根据配置动态生成（主要是替换macro）

给Visitor用邮件发送Transcript 时:

1. 要用agent 的名字替换掉邮件subject 中的macro :｛agent｝
   * 1. 数据结构的修改

无

* + 1. 处理逻辑

聊天结束时给指定邮箱发送transcript

DbAccess 的AttachToTicketAndSendEmailWhenSavingChat方法要做如下修改:

1. codeplan 的IfEnableChatWindowSendTranscriptEmail为true并且收件人地址不为空时才字段发送邮件
2. 邮件的subject 要根据codeplan的ChatWindowSendTranscriptEmailSubject

字段来动态生成, 需要替换掉其中的macro

发给visitor的邮件

1. Subject 中的 macro :{agent} 用Operator1Name来替换.

*备注: 一个聊天可能会有多个Operator，但是如果把所有的Operator Name都拿出来替换｛agent｝(Operator Name 之间用[,]隔开) 可能会让subject 有语法错误*

* + 1. 数据库修改

参看portal端的设计文档

##### Max-on 延长单个请求的延时时间

* + 1. 整体描述

现在单个请求的延时如果超过2秒就会认为服务器有问题, 导致副服务器判断主服务器状态过于敏感，很容易误切换, 为减少误切换，把超时时间设置为6秒

* + 1. 数据结构的修改

ModeratorHandler 增加属性

1. WebClient[] \_webClients *//里面有3个WebClient*
2. Short \_currentWebClientIndex *//标记当前被轮询到的WebClient的Index*
   * 1. 处理逻辑

2分钟时间间隔不变, ModeratorHandler 内置3个Webclient, 在Monitor内轮询调用DownloadDataAsync() 方法, 每个WebClient 下一次被轮询到的时间间隔刚好是6分钟，如果此时IsBusy 为true， 就调用CancelAsync() 方法

* + 1. 数据库修改

无

##### Max-on修改判断数据库是否可用的逻辑

* + 1. 整体描述

检查数据库是否正常工作的CheckDb()函数 只包含对数据库的读操作, 当数据库的磁盘空间满时，数据库不可用, 但是可以正常进行读取操作, 因此在CheckDb() 需要增加一个写操作

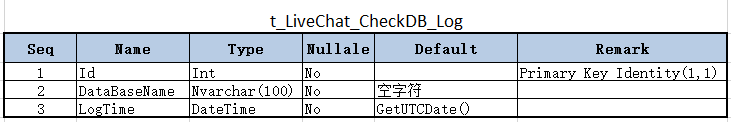
* + 1. 数据结构的修改

无

* + 1. 处理逻辑

读操作结束后，向General 库的t\_LiveChat\_CheckDB\_Log 表插入一条数据, 只有读写操作均成功时才认为数据库是可用的

* + 1. 数据库修改

在General 库加新表t\_LiveChat\_CheckDB\_Log

##### 压缩WebService Soap 减少网络带宽

* + 1. 整体描述

现在数据库同步时传输的数据比较大，希望能压缩或差量同步来减少对网络的压力。

解决方案就是给WebService Soap 增加压缩功能, 传送数据前要压缩, 接收数据后要解压缩

* + 1. 数据结构的修改

无

* + 1. 处理逻辑

1. 使用soap扩展, 新建一个继承于SoapExtensionAttribute 的Attribute

CompressionSoapExtensionAttribute.

1. 重写该属性的ProcessMessage 方法，在Serialize 以后使用GZipStream 对数据进行压缩

在Deserialize 以前使用GZipStream 对数据进解压

1. 在需要压缩数据的方法前加上该属性
   * 1. 数据库修改

无

##### Custom variables displayed in offline messages

* + 1. 整体描述

要把Custom variable 和对应的 offline message 关联并保存在数据库

* + 1. 数据结构的修改

1. PendingOfflineMessage 增加 CustomVariableResult[] CustomVariables 属性 （构造函数也要增加）
2. SessionOfflineMessage增加 CustomVariableResult[] CustomVariables 属性（构造函数也要增加）
   * 1. 处理逻辑
3. 提交offlineMessage 时把 CurrentVisitor.CustomVariableResult的值和offlineMessage 关联起来，保存到数据库，具体需要增加CustomVariables 参数的方法有:
4. PendingOfflineMessage 的SubmitOfflineMessage 方法
5. OfflineMessages 的AddOfflineMessage 方法
6. OfflineMessageAccess 的AddOfflineMessage 方法
7. 获取SessionOfflineMessage 时可以通过CustomVariables 属性的值得到相关的自定义变量
   * 1. 数据库修改

无

##### Monitor All Chats & Monitor My Departments’ Chats

* + 1. 整体描述

对Monitor Chat 进行更具体的控制, 控制的方法是引入2个权限入口: Monitor

All Chats, Monitor My Departments’ Chats.

**Monitor All Chats**

可以Monitor 本企业所有的Chat, 拥有此权限的Agents 也同时拥有Monitor My Departments’ Chat 权限.

**Monitor My Departments’ Chats**

只能Monitor 自己所属部门的chat

* + 1. 数据结构的修改

1. EnumLiveChatPermission中增加MonitorMyDepartmentsChats = 1<<30, Monitor All Chats沿用MonitorChat项，项名改为MonitorAllChats
2. EnumOperatorPermission中在Live Chat部分增加LiveChat\_MonitorMyDepartmentsChats
3. EnumValueChange的GetPermissionValue中增加LiveChat\_MonitorMyDepartmentsChats转为MonitorMyDepartmentsChats的case
   * 1. 处理逻辑

ChatServer:

ChatEngineWrap中OperatorAddMessage方法如果是enumOperatorMonitorChat时，判断这个operator是否有monitor权限

1. 如果有MonitorAllChats权限，则直接通过
2. 如果没有MonitorAllChats权限，再判断是否有MonitorMyDepartmentsChats
3. 如果没有MonitorMyDepartmentsChats权限，直接抛异常
4. 有MonitorMyDepartmentsChats权限，并且这个聊天是这个operator所在部门，则通过
5. 有MonitorMyDepartmentsChats权限，但这个聊天不属于这个operator所在部门，直接抛异常

VM:

1. 获取CurrentOperator的LiveChatPermission 字段,根据LiveChatPermission 字段来判定operator 是否有Monitor All Chats/Monitor My Departments’s Chats
2. 在Visitor 列表里，如果当前agent 对某聊天有Monitor 权限就显示 [monitor] 按钮，否则就不显示
   * 1. 数据库修改

无