



技術手冊

HomeMeeting 整合工具 3.23.9 版

2017 一月

免責聲明；不附隨保證

這份資訊以及其它所有文件（印刷或電子檔），只供參考之用。雖然本公司已盡力確保這份資訊的完整性與準確性，這份資訊以及其它所有文件係以“現況”提供，但 HOMEMEETING INC. 不對任何明示或默示的擔保及條件負責，其中包括但不限於適售性、適合某特定用途等情形。HOMEMEETING INC. 將不會對使用這份資訊以及其它所有文件或與之相關而引起任何損害，包括但不限於直接、間接、衍生或附帶的損害承擔任何賠償。在這份資訊以及其它所有文件中，HOMEMEETING INC.（或其供應商或授權商）做任何保證或陳述，或修改使用這個軟體適用授權合約的條款及條件，有任何未盡事項與此相反，也不應具有效力。如本免責聲明及版權告示的中、英文兩個版本有任何抵觸或不相符之處，須以英文版本為準。

版權

根據此法例及權利，除了在憑證中或管制使用該軟體的許可協議中所描述的方式之外，在未經 HomeMeeting Inc. 書面同意下，這份文件或軟體之全部或部份均不可被複印、影印、重製、翻譯或轉換成任何電子檔或機器可判讀的格式。

© Copyright 2013 HomeMeeting Inc
15357 NE 90th Street
Redmond, WA98052

版權所有。在美國印刷

目錄

壹、 背景	1
貳、 傳遞驗證	2
參、 第三方應用程式與 MCU 之間的介面	5
3.1 請求服務的 XML 格式	5
3.2 MCU 組態檔	15
3.3 錄影檔目錄結構	16
3.4 會議室主人會議狀態檔	16
3.5 錄影檔狀態檔	20
3.6 與會者資訊檔	21
3.7 準備模式	22
3.8 伺服器備援	24
3.9 伺服器叢集與負載平衡	25
3.10 密碼驗證	27
3.11 第三方應用程式規則	28
3.12 錄影檔保管	29
3.13 一次性 MCU Relay	30
肆、 JoinNet Messenger	31
4.1 MCU 的組態配置	31
4.2 獲取 MCU 的 IP 與 Port	31
4.3 MCU 與第三方應用程式之間的互動 (由 MCU 發起的互動)	31
4.4 請求列表	32
4.5 給第三方應用程式使用的線上資訊	48
4.6 Messenger 負載資訊	49
4.7 在 Messenger 伺服器與會議伺服器之間	49
4.8 第三方應用程式與 Messenger 伺服器之間的互動 (由第三方應用程式發起的互動)	50
4.9 共享與取消共享錄影檔	50
4.10 MCU 叢集	51

壹、背景

HomeMeeting 整合工具 (以下簡稱 HIT) 是為了讓第三方應用程式與 HomeMeeting 伺服器可以靈活彈性地整合所設計的。

對第三方應用程式來說，HomeMeeting 伺服器 (以下簡稱 MCU) 單純只是個會議服務提供者，從受信任的第三方應用程式接收會議請求，本身不直接進行任何使用者的身份驗證。第三方應用程式可以是個網頁伺服器或是任何與終端使用者互動的介面。第三方應用程式負責維護使用者資料庫、驗證使用者身分、建立會議請求給 MCU 伺服器，以及提供使用介面給其終端使用者。

一旦通過第三方應用程式的驗證，使用者可以請求召開會議、播放錄影檔或共享錄影檔等服務。例如，當一位使用者請求召開會議，第三方應用程式接著產生一個新的 jnj 檔，允許使用者利用這個 jnj 檔來連接到 MCU。使用者資訊以及請求服務的內容已經寫入在這個 jnj 檔中。MCU 將會檢查 jnj 檔的有效期，只有當 jnj 檔還在有效期內，MCU 才會接受 jnj 檔並進行處理 (預設，jnj 檔在產生後的有效期為 5 分鐘)。請注意，將 MCU 與第三方應用程式兩者的時間同步是非常重要的，否則在檢查有效期時將可能發生 jnj 過期的問題。同步 MCU 與第三方應用程式兩者時間的機制並不在此文件範圍內，請自行參考相關文章。

下列是由第三方應用程式所產生的 jnj 範例：

```
[general]
domain = HomeMeeting
codetype = 13
ip = 192.168.1.63
action = 0
UserInfo = key_web_192.168.1.63_0000|Bq6=|NOyO1+r1iO7R|nrSn9048=
```

這種類型的 jnj 檔很容易受到被動式攻擊，如果攻擊者可以攔截到藉由網路傳輸的 jnj 檔，惡意攻擊者可以假冒成真正的使用者。要避免被動式攻擊，建議使用如 https 連線的安全傳輸來傳輸 jnj 檔。請注意，在 jnj 檔中所顯示的 'action' 項目代表請求服務為召開會議 (0) 或播放錄影檔 (1)；'codetype' 項目為 13，是所有 HIT 請求服務的預定值。

HIT 也支援終端使用者與 MCU 之間直接進行驗證。當使用直接驗證時，jnj 檔將不會檢查有效期，所以不需要確認 MCU 伺服器與第三方應用程式之間的時間是否同步。直接驗證也可以避免被動式攻擊，因為使用者所輸入的密碼是透過安全的 JoinNet-MCU 頻道進行傳輸。密碼驗證就是直接認證的範例之一。

貳、傳遞驗證

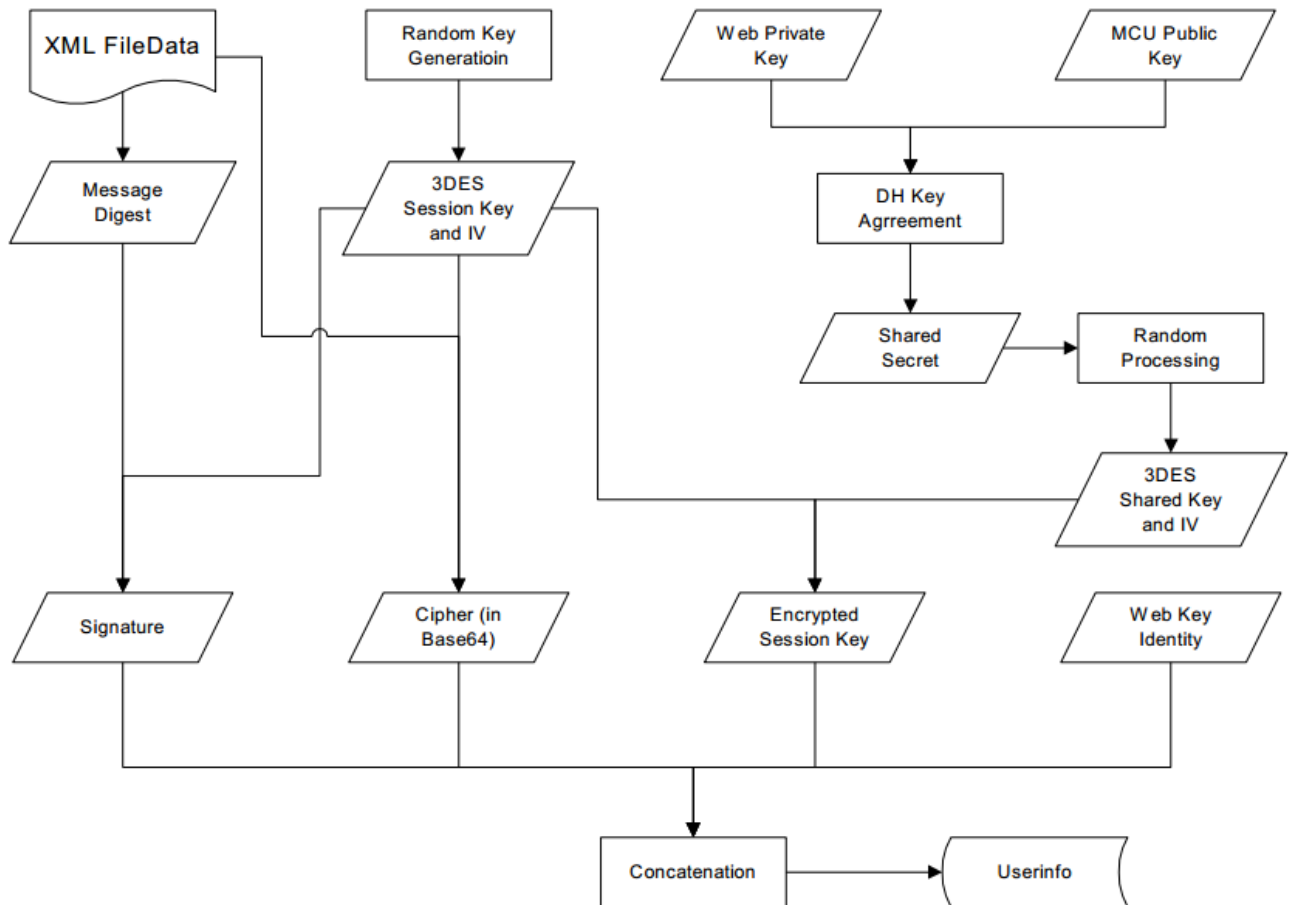
HIT 採用公鑰基礎架構從第三方應用程式來傳遞驗證給 MCU。傳遞驗證與請求服務是使用特定的 Diffie-Hellman 金鑰交換演算法。對於 DH 金鑰交換演算法，固定的參數是：

- Prime (十六進制)：

```
FFFFFFFFFFFFFFFFC90FDAA22168C234C4C6628B80DC1CD129024E088A67CC740  
20BBEA63B139B22514A08798E3404DDEF9519B3CD3A431B302B0A6DF25F14374F  
E1356D6D51C245E485B576625E7EC6F44C42E9A637ED6B0BFF5CB6F406B7EDEE3  
86BFB5A899FA5AE9F24117C4B1FE649286651ECE45B3DC2007CB8A163BF0598DA  
48361C55D39A69163FA8FD24CF5F83655D23DCA3AD961C62F356208552BB9ED52  
9077096966D670C354E4ABC9804F1746C08CA18217C32905E462E36CE3BE39E77  
2C180E86039B2783A2EC07A28FB5C55DF06F4C52C9DE2BCBF6955817183995497  
CEA956AE515D2261898FA051015728E5A8AACAA68FFFFFFFFFFFFFFFF (這是一個  
2048 位元質數。MCU 3.23.0 (含) 以下的版本，採用 1024 位元質數)
```

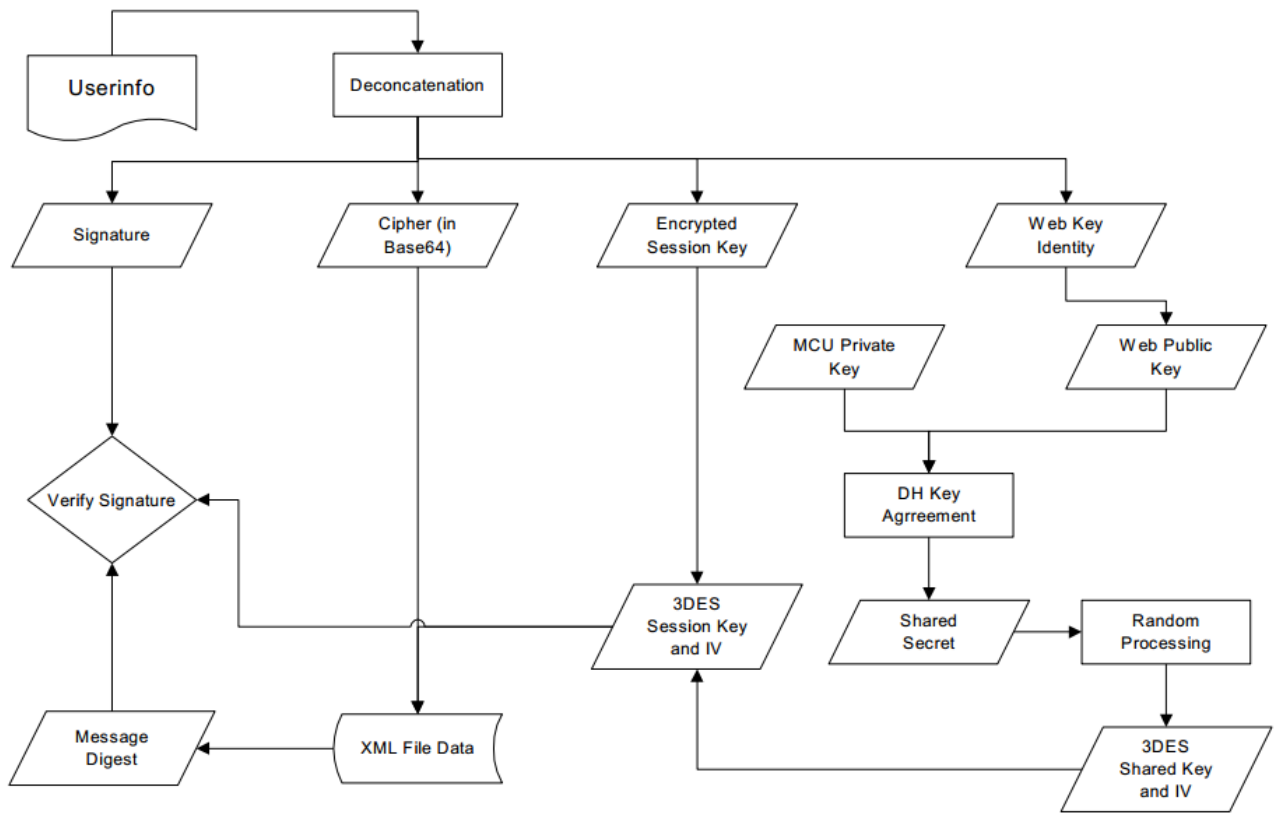
- Generator: 2

MCU 與第三方應用程式兩者均持有一對 DH 金鑰，用來加密與驗證承載在 jnj 檔中的請求服務。第三方應用程式首先產生一個 XML 檔，內容包含使用者資訊與請求服務；接著這個 XML 檔的處理流程如圖一所示：



圖一：產生使用者資訊

使用者資訊將會做為一個項目並寫入在 jnj 檔中。當終端使用者開啟這個 jnj 檔時，使用者資訊將被傳遞至 MCU。MCU 會藉由包含在使用者資訊中的網頁金鑰身分來找到網頁公鑰檔。如果找不到網頁公鑰檔，請求服務將被拒絕。MCU 解密與驗證在使用者資訊中 XML 檔資料的處理流程如圖二所示：



圖二：MCU 解密與驗證使用者資訊

參、 第三方應用程式與 MCU 之間的介面

3.1 請求服務的 XML 格式

終端使用者的請求服務會儲存成 XML 格式。

```
<? xml version="1.0" encoding="utf-8" ?>
<jnj>
  <owner id="ownerid" diskquota="1000" maxoutconnection="50"
email="owner@email.com">ownername</owner>
  <meetingid>testing</meetingid>
  <timestamp>1019077830</timestamp>
  <password>Base64 encoded SHA hash of password</password>
  <guest id="guestid" email="guest@email.com" invited="1"
ticket="0">guestname</guest>
  <group diskquota="1000" maxoutconnection="50">HomeMeeting</group>
  <command recording="1" exclusive_recording="0" autoextension="0"
duration="60" default_joint_browsing_page="" all_questioner="0"
preparationmode="1" bandwidth="64000" total_av_bandwidth="256000"
max_message_duration="10" joinnet_auto_update_url=""
joinnet_holder_video="1" audio_codec_preference=""
video_codec_preference="" file="recording_file_name">meeting
  <guaranteed invited="10" uninited="3">1</guaranteed>
  <transferslide userid="userid" groupid="groupid"
file="recording_file_name">meetingid</transferslide>
  <continuemeeing userid="userid" groupid="groupid"
file="recording_file_name">meetingid</continuemeeing>
  </command>
  <meetingtitle>testing title</meetingtitle>
</jnj>
```

圖三：請求服務範例

圖三是一個請求服務的範例。在 XML 檔中有四項必要的項目：“owner”、“meetingid”、“timestamp”以及主持人的“id”。剩下的其它元素以及參數則為非必要項目。

所有的字串元素必須採用 utf-8 編碼，而每一項字串元素的長度限制如下所示：userid、username、groupid、meetingid 以及 email 不得超過 128 位元組；meeting title、filename 以及 default joint browsing page 不得超過 512 位元組；password 不得超過 40 位元組。

當請求服務的目的是召開會議時，只有實際建立會議的第一次請求服務會設定會議相關的屬性。接著加入相同會議的所有請求服務將不會影響現行會議的屬性。這項規則有一些例外情況。例外情況如下。

- 下列元素並不會被儲存為會議屬性：`<owner maxoutconnection>`、`<password>`、`<timestamp>`、`<group maxoutconnection>`。
- 當請求服務者為位訪客時，例外情況包括：`<guest>`。
- 當請求服務者是位主持人時，例外情況包括：`<owner>`、`<owner email>`。
- 當請求服務者是位從準備模式切換至一般會議模式的主持人時，例外情況包括：`<owner>`、`<owner email>`、`<group>`、`<meetingtitle>`、`<command autoextension>`、`<command duration>`、`<command guaranteed>`、`<command guaranteed invited>`、`<command guaranteed uninvited>`。

每一項元素的用法與規範說明如下。

1. “owner” 元素 [必要項目]：“owner” 是指定主持人的使用者名稱。這個名稱將會顯示在 JoinNet 的控制臺中。

- “id” 參數 [必要項目]：“id” 是指定主持人的帳號。這個主持人的帳號在第三方應用程式中必須為獨一無二的值。如果請求服務是召開會議請求，在會議中所產生的錄影檔會被計算在該主持人的磁碟用量中，所有與會者的網路連線也會被計算在該主持人的與會者人數中。
- “diskquota” 參數 [非必要項目，預設值為 50，可在 configm.ini 檔中調整預設值]：“diskquota” 是指定主持人儲存錄影檔的最高磁碟用量（單位為 MB）。當請求服務為 “meeting” 且會議啟動錄影，主持人錄影檔產生的磁碟用量將會核對 “diskquota”。當磁碟用量超過 “diskquota” 時，召開會議請求將被拒絕。

磁碟用量的檢查只會在會議開始前進行。當任一會議結束且產生新錄影檔時，磁碟用量加上新錄影檔的大小可能會超過 “diskquota”。然而，即使加上新錄影檔之後的磁碟用量大小超過 “diskquota”，新錄影檔還是會被存檔。這也表示使用者的磁碟用量可能會超過他的 “diskquota”。此參數的預設值是由 MCU 的組態檔 configm.ini 中 (default_disk_quota) 所指定。如果在 configm.ini 中未賦予 default_disk_quota 數值，則預設值為 50MB。

第三方應用程式可以將 “diskquota” 指定為 0，這是通知 MCU 不要執行 diskquota 檢查。在這種情況下，則由第三方應用程式負責執行 diskquota 檢查。在 MCU 的組態檔中還有另外一項配置—“grace disk quota”。這項配置值數值為百分比。例如，20 代表寬容值為 20%。在此配置啟用狀態下，當磁碟用量超過 “diskquota”，會再納入寬容值計算。

- “maxoutconnection” 參數 [非必要項目，預設值為 10，可在 config.ini 檔中調整預設值]：“maxoutconnection” 是指定計算在主持人的最大外部連線人數。外部連線泛指連線到 MCU 的 IP 不在 MCU 授權檔中 IP range 以內的 IP。如果在授權檔中未指定 IP range，所有連線將被認定為外部連線。計算在主持人的連線包括：(a) 主持人自己本身的連線（包括召開會議與播放錄影檔）、(b) 在主持人會議中的所有與會者連線、(c) 在主持人會議中所有被撥打的 SIP 電話連線。當現行所有計算在主持人的外部連線數量超過 “maxoutconnection” 時，請求服務將被拒絕。如果未使用此參數，則 “maxoutconnection” 的預設值是由 MCU 的組態檔 config.ini 中 (default_max_outconnection) 所指定。如果在 config.ini 中也未賦予 default_max_outconnection 數值，則預設值為 10。
- “email” 參數 [非必要項目]：“email” 是指定主持人的電子郵件地址。當 MCU 啟用支援電子郵件通知時，在下列情況下 MCU 將會寄送電子郵件到這個 “email” 地址：(a) 主持人的磁碟用量超過 diskquota 時。(在會議結束且未併入寬容值計算就已經超過時，就會寄出電子郵件)、(b) 訪客留言給主持人時。
請注意，由 MCU 所寄出的電子郵件是非常初級的內容，如果可能的話，強烈建議交由第三方應用程式來寄出這類的警示電子郵件給主持人。當這類的事件發生時，MCU 會藉由 Messenger 介面來通知第三方應用程式。有關 Messenger 介面的說明將在後續的章節說明。

2. “meetingid” 元素 [必要項目]：“meetingid” 用在兩個方面。一是將錄影檔儲存在 “meetingid” 的子目錄，好讓第三方應用程式可以依據 “meetingid” 來找到錄影檔的儲存路徑。二是使用 “meetingid” 來識別會議的唯一值。

由於 “meetingid” 是會議的識別碼，所以在該主持人的帳號消失之前必須是唯一值。第三方應用程式必須確保為每一位使用者所產生的 jnj 檔中的 “meetingid” 是唯一值。

“meetingid” 的內容最高支援至 128 位元組（不包括結尾的空值），所以要使其內容不重複且唯一並不是件難事。簡單的作法是：(a) 使用一個整數，並在每一場會議產生 jnj 檔之後將整數加一，等到達整數的最大值時再繞回，或是 (b) 使用系統產生的 GUID—這可能會導致在錄影檔目錄/文件結構中產生非常長的目錄名稱。

請注意：當請求服務為 “checkmessage” 時，不使用 “meetingid”。

3. “timestamp” 元素 [必要項目]：“timestamp” 是用來檢查請求服務的有效期，如果在 “timestamp” 中所包含的時間跟 MCU 自身時間的誤差在**五分鐘**以內，請求服務就被視為有效，超過五分鐘將被拒絕。

當 XML 檔中出現 “password” 時，“timestamp” 則是指示 “password” 的到期時間（更多說明請見 3.10 章節）。

4. “password” 元素 [非必要項目]：“password” 包含以 base64 加密進行安全雜湊演算法後的密碼。MCU 將會使用這些資訊來驗證終端使用者，例如，如果密碼為 “testpassword”， “password” 元素的值則為 “i7YRj4/Wk1rQh2o740pxfTJwj/0=”。

當 “password” 元素出現時，“timestamp” 則是指示 “password” 的到期時間。

5. “guest” 元素 [非必要項目]：“guest” 是指定訪客的名稱。如果 XML 檔中出現此元素時，請求服務者將被視為**訪客**；否則請求服務者將被視為**主持人**。

一般情況下，訪客無法開啟會議室。只有主持人可以召開會議，即開啟即時型會議室。然而，受邀請的訪客則是可以開啟預約型會議室，即使在主持人遲到的情況下。即時型會議室與預約型會議室之間的不同點將在後續再進行說明。當主持人不在線上而訪客選擇留言給主持人時，訪客實際上是開啟了一間以主持人為名的即時型會議來進行留言。

請注意：當指令為 ‘playback’ 或 ‘download’ 時，訪客的 ‘id’ 與 ‘name’ 資訊則是代表播放者資訊或是下載檔案的使用者資訊。

- “id” 參數 [非必要項目]：“id” 是指定訪客的帳號。在 XML 檔中帶有 “id” 的訪客將被視為可識別的訪客；除此之外，因為使用者名稱無法用來識別訪客，沒有 “id” 的訪客將被視為匿名者。由於 “id” 是用來識別訪客，所以應該跟主持人的 “id” 一樣是唯一值。

在同一場會議中，如果兩位不同使用者名稱的訪客擁有相同的 “id”，他們將被視為**同**

一人，只有其中一人可以參加會議。(為了避免混淆，第三方應用程式應該防止這種情況發生) 在另外一方面，如果有兩位擁有相同使用者名稱的訪客沒有指定 "id"，他們將被視為**不同人**，兩位都可以參加會議。

只有帶有 "id" 的訪客能夠被主持人永久退出會議室。

- "email" 參數 [非必要項目]："email" 是指定訪客的電子郵件地址。只有帶有 "id" 的訪客能夠指定 "email" 資訊。
- "invited" 參數 [非必要項目，預設值為 "0"]："invited" 是指定訪客是否為受邀者。如果訪客為受邀者 ("invited" 等於 1)，訪客可以自由進入主持人的會議室而不用經過主持人的授權允許。除此之外，沒有受到邀請的訪客要進入主持人的會議室必須要經過主持人的授權允許，方可進入。如果主持人在 JoinNet 的設定中勾選 "請勿打擾"，未受到邀請的訪客在主持人取消勾選之前將再也無法拜訪主持人的會議室。請注意，"id" 與 "invited" 兩項參數是各自獨立的。所以總共有四種訪客：(a) 可識別且受邀請者、(b) 可識別但未受邀請者、(c) 匿名且未受邀請者、(d) 匿名但受邀請者—不建議這種設定。

- "ticket" 參數 [非必要項目，預設值為 "0"]："ticket" 是指定訪客是否為門票持有者。MCU 授權檔中必須支援門票型會議室才能接受門票持有者的請求服務。"ticket" 僅能用於預約型會議室。在主持人尚未開啟會議室前，即使門票持有者的 "invited" 為受邀者也無法參加活動。當主持人建立門票型會議室時，門票持有者可以自由進入主持人的會議室而不用經過主持人的授權允許，如同受到邀請的訪客一樣；此外，門票持有者的連線會被計算在受邀者的連線人數。

門票持有者在會議期間**無法**傳送視訊。門票持有者的使用者資訊只會顯示在會議室主人的控制臺中，也就是說，其他與會者在控制臺中將不會看見門票持有者。在門票型會議室中，JoinNet 的自動允許發言功能將無法使用。

請注意，門票持有者必須有 "id" 參數，也就是說門票持有者必須是可識別的。

6. "group" 元素 [非必要項目]："group" 是指定**主持人**的群組帳號。當 XML 檔中出現此元素時，與群組相關的限制也會套用在請求服務中，且錄影檔也會被寫入至與 "group" 相對應的子目錄中。

- "diskquota" 參數 [非必要項目，預設值為 50，可在 config.ini 檔中調整預設

值]：“diskquota”是指定群組儲存錄影檔的最高磁碟用量（單位為 MB）。當請求服務為 “meeting” 且會議啟動錄影，群組錄影檔產生的磁碟用量將會核對 “diskquota”。當磁碟用量超過 “diskquota” 時，召開會議請求將被拒絕。更多說明請見 “owner” 元素章節。

- “maxoutconnection” 參數 [非必要項目，預設值為 10]：“maxoutconnection”是指定最大外部連線人數。當現行所有計算在群組的外部連線數量超過 “maxoutconnection” 時，請求服務將被拒絕。

如果未使用此參數，則 “maxoutconnection” 的預設值是由 MCU 的組態檔 configm.ini 中 (default_max_group_outconnection) 所指定。如果在 configm.ini 中也未賦予 default_max_group_outconnection 數值，則預設值為 10。

7. “command” 元素 [非必要項目，預設值為 “meeting”]：“command” 是指定請求服務類型。總共有七種請求服務：(a) meeting 召開會議、(b) playback 播放錄影檔、(c) download 下載錄影檔、(d) delete 刪除錄影檔、(e) checkmessage 查詢留言、(f) talktovistor 與訪客交談以及 (g) leavemessage 留言。

請注意，當請求服務為 ‘delete’ 或是 ‘checkmessage’ 時，錄影檔才可以被刪除。如果第三方應用程式擁有自己的錄影檔管理機制，且不讓任何終端使用者可以直接刪除錄影檔，第三方應用程式就不要產生這兩種類型的 jnj 檔。

‘talktovistor’ 是種特別的指令，可以讓主持人（可識別的 ownerid）與正在留言（可識別的 meetingid）給主持人的訪客進行交談。產生這種指令有幾項限制：主持人不能有正在進行中的會議。

‘leavemessage’ 指令（僅產生給訪客使用）不管會議室主人是否開啟他的會議室，訪客將會留言給會議室主人如同 ‘meeting’ 一樣。

- “recording” 參數 [非必要項目，預設值為 “1”]：“recording” 是指定會議是否進行錄影。如果指令非 “meeting” 此項參數將被忽略。當請求服務者為訪客而此參數設定為 “0” 時，如果會議室主人不在線上，請求服務者將無法留言。
- “autoextension” 參數 [非必要項目，預設值為 “1”]：“autoextension” 是指定當預約型會議室在會議時間超過 “duration” 之後，是否自動延長時間。如果這項參數

設定為“0”時，MCU 將在會議時間超過“duration”時終止會議。如果這項參數設定為“1”時，只要有任何一位與會者還在會議中，即使會議時間超過“duration”，會議還是會繼續進行。如果指令非“meeting”或是“duration”參數沒有設置，此項參數將被忽略。

- “duration” 參數 [非必要項目]：“duration”是指定會議的持續時間。“duration”是會議的持續時間，以分鐘為單位，其數字不得超過 600 (10 小時)。任何持續時間大於或等於 600 將被減為 599。如果指令非“meeting”此項參數將被忽略。如果此項參數出現，且其值不為“0”，會議將被視為預約型會議；否則，會議將被視為即時型會議。表一說明即時型會議與預約型會議的差異：

情境	即時型會議	預約型會議
當會議室主人尚未開啟會議室，有受到邀請的訪客請求會議服務時	訪客可以進行留言	訪客開啟會議室開始參加會議 (在此情況下，訪客將擁有主持權)
當會議室主人尚未開啟會議室，有未受到邀請的訪客請求會議服務時	訪客可以進行留言	訪客會收到“這個會議無法在預約的時間以外開始或這個會議已經結束”的錯誤訊息
會議何時被終止？	<ol style="list-style-type: none"> 1. 當主持人關閉 JoinNet 且明確地終止會議時 2. 當所有與會者離開會議室且沒有人在會議室達一分鐘時 	<ol style="list-style-type: none"> 1. 當主持人關閉 JoinNet 且明確地終止會議時 2. 當 autoextension 設定為“0”，且會議時間超過“duration”時 3. 當 autoextension 設定為“1”，在會議時間超過“duration”之後，所有與會者離開會議室且沒有人在會議室達一分鐘時
所有與會者離開會議室，會議室沒有人	MCU 伺服器會將會議終止。若是有任何與會者返回，他將會進入新的會議室。	會議將持續進行直到會議時間超過“duration”為止。若是有任何與會者返回，他將會進入原會議室。
如果兩位主持人使用相同的 meetingid (請注意，這是個不應該發生的錯誤情況)	這兩位主持人將開啟各自的會議室，會議將被錄影儲存在單獨的目錄下	第一位主持人開啟會議室，而第二位主持人則會收到“Internal error”的錯誤訊息
如果同一位主持人請求兩個不同 meetingid 的會議服務 (這種情況可能會發生在從不同電腦啟動時)	第一位請求者將會開啟會議室成為主持人，第二位請求者將會進入相同的會議室，並將第一位請求者踢出	產生兩間會議室

表一：即時型會議與預約型會議的差異

- “default_joint_browsing_page” 參數 [非必要項目] :
“default_joint_browsing_page” 是指定會議中共同瀏覽網頁的首頁。如果指令非 “meeting” 此項參數將被忽略。
- “exclusive_recording” 參數 [非必要項目，預設值為 “0”] :
“exclusive_recording” 是指定當錄影檔資料夾中已經有舊錄影檔時是否可以啟動會議。如果指令非 “meeting” 此項參數將被忽略。當設定此參數且錄影檔資料夾中有舊錄影檔時，會議將無法啟動。
- “all_questioner” 參數 [非必要項目]: “all_questioner” 是指定特定類型的會議，即所有與會者同時也是發言者。一間會議室可以支援的發言者人數是由 HomeMeeting 的授權以及 MCU 的 config.ini 檔所控制的。如果現行發言者人數達到它的門檻，即使有其它設置或預留連線，將沒有人可以再進入會議室。
- “preparationmode” 參數 [非必要項目]: “preparationmode” 是指定預約型會議的準備模式。即時型會議此項參數將被忽略 (即會議的持續時間為 0)。此項參數的詳細說明請見 “準備模式” 章節。
- “bandwidth” 參數 [非必要項目]: “bandwidth” 是指定使用者可以用在視訊傳送速率的最高頻寬。單位為 bps。
- “total_av_bandwidth” 參數 [非必要項目]: “total_av_bandwidth” 是指定一間會議室可以被分配用在視訊及語音上的最高頻寬。當接收到此資訊，每一個 JoinNet 將會根據同時發言者的人數調整其視訊傳送速率。單位為 bps。請注意，只有 jnkernel 1.23 以上版本的 JoinNet 可以接收此資訊。
- “max_message_duration” 參數 [非必要項目]: “max_message_duration” 是指定訪客可以進行留言的最長持續時間。當剩下 5 分鐘時，訪客將會收到由 MCU 伺服器所傳送會議即將終止的警告訊息。
- “joinnet_auto_update_url” 參數 [非必要項目]: “joinnet_auto_update_url” 是告知 JoinNet 檢查自動更新的位置。此項參數的值是網址，長度不得超過 512 位元組 (包括結尾的空值)。請注意，只有 jnkernel 1.23 以上版本的 JoinNet 可以接收此資訊。

- “joinnet_holder_video” 參數 [非必要項目]：“joinnet_holder_video” 設定為 1，JoinNet 預設將在主影像視窗顯示主持權持有者的視訊。否則，預設將顯示發言者的視訊。請注意，只有 jnkernel 1.23 以上版本的 JoinNet 可以接收此資訊。
- “audio_codec_preference” 參數 [非必要項目]：“audio_codec_preference” 是告知 JoinNet 這次會議使用語音編碼的偏好設定。截至目前為止，支援的語音編碼有：dflt、ilbc、g711、g762 和 opus。這些語音編碼的預設頻寬（不含其它雜項）為：dflt (6.4kbps)、ilbc (13.5kbps)、g711 (64kbps)、g726 (16kbps)、opus (可配置)。請注意，只有 jnkernel 1.23 以上版本的 JoinNet 可以接收此資訊。
- “video_codec_preference” 參數 [非必要項目]：“video_codec_preference” 是告知 JoinNet 這次會議使用視訊編碼的偏好設定。截至目前為止，支援的視訊編碼有：dflt、hm264、f264、mjpg 和 vpx。請注意，只有 jnkernel 1.23 以上版本的 JoinNet 可以接收此資訊。
- “file” 參數 [非必要項目]：“file” 是指定錄影檔的檔案名稱（不包括副檔名 “.jnr”）。此參數只對指令 “playback”、“download”、“delete” 有用。通常，“meetingid” 是用來識別要進行播放、下載或刪除的錄影檔。然而，一個 “meetingid” 可能會有多個錄影檔。此項參數是用來指定唯一的錄影檔。
如果此項參數未出現，且單一 “meetingid” 擁有多個錄影檔時，則擁有最新建立日期的錄影檔將被選取。
- “transferslide” 子元素 [非必要項目]：“transferslide” 是指定 meetingid 的錄影檔中哪些投影片將被傳送到會議。當指令為 “meeting” 時才會使用此項子元素。“transferslide” 有三項參數：“userid”、“groupid” 和 “file”。“userid”、“groupid” 和 “meetingid” 在一起可以定義一個錄影檔，即屬於 “userid”、“groupid” 為了 “meetingid” 所使用的錄影檔。如果使用者不屬於任何群組，“groupid” 是非必要的。如果有多個錄影檔供一組特定的 [“userid”、“groupid”、“meetingid”] 使用，最新的一個錄影檔將供傳送投影片使用。另一項非必要的參數 “file” 則可以用來指定唯一的錄影檔。
- “guaranteed” 子元素 [非必要項目]：“guaranteed” 是指定是否預約型會議可以

使用保障的資源。如果此項子元素設定其值為“1”，有兩項參數“invited”和“uninvited”定義在會議中保留多少位受到邀請與未受到邀請的訪客。如果會議非預約型會議，此項子元素將被忽略。

- “continuemeeeting”子元素 [非必要項目]：“continuemeeeting”是指定這次會議是接續那一個 meetingid 的錄影檔。當指令為“meeting”時才會使用此項子元素。當指定此子元素時，則在錄影檔中所錄影的資訊將被重新載入至新啟動的會議，會議將從錄影檔結尾處繼續進行。

“continuemeeeting”有三項參數：“userid”、“groupid”和“file”。“userid”、“groupid”和“meetingid”在一起可以定義一個錄影檔，即屬於“userid”、“groupid”為了“meetingid”所使用的錄影檔。如果使用者不屬於任何群組，“groupid”是非必要的。如果有多個錄影檔供一組特定的 [“userid”、“groupid”、“meetingid”] 使用，最新的一個錄影檔將供傳送投影片使用。另一項非必要的參數“file”則可以用來指定唯一的錄影檔。

8. “meetingtitle”元素 [非必要項目]：這項元素只用來寫入在每個錄影檔相對應狀態的 XML 檔裡面。它的用途與會議無關。此項元素的用法稍後再做說明。

3.2 MCU 組態檔

MCU 必須正確地配置才能接收由第三方應用程式所產生的請求服務。

傳遞驗證所使用的 DH 金鑰對可以由一個獨立的程式產生。在產生特定的金鑰對之後，將金鑰對複製到 MCU 目錄中 (MCU 執行檔所在位置的目錄)，接著在 MCU 的組態檔中設定下列項目：

```
mcu_cluster_dh_key=mcu_key_identity
```

```
passphrase=passphrase_of_mcu_priv_key
```

當執行金鑰對產生程式時，會將通關密碼寫入在“passphrase.txt”檔案中。

若要接收來自特定第三方應用程式的請求服務，只需要將第三方應用程式的公鑰放到 MCU 的目錄中即可。將多組公鑰放到單一 MCU 的目錄中，則該 MCU 可以接收來自多個第三方應用程式的請求服務。

若要拒絕某個第三方應用程式的請求服務，只須從 MCU 目錄中刪除該第三方應用程式的公鑰即可。

3.3 錄影檔目錄結構

在第三方應用程式結構的錄影檔是採階層系統進行儲存。假設錄影檔跟目錄的配置是“~root”。如果在使用者資訊 (XML 格式檔) 中沒有“group”，那麼錄影檔是儲存成：

~root/_user/ownerid/meetingid/_recording_ip_date_sessionindex.jnr

如果在使用者資訊 (XML 格式檔) 中有“group”，那麼錄影檔是儲存成：

~root/_group/groupid/ownerid/meetingid/_recording_ip_date_sessionindex.jnr

在某個“groupid”目錄底下的所有錄影檔大小（包括所有子目錄）將被定義為該群組的總磁碟用量。

~root/_group/groupid

在某個“ownerid”目錄底下的所有錄影檔大小（包括所有子目錄）將被定義為該主持人的總磁碟用量。

~root/_user/ownerid 與 ~root/_group/*/ownerid

在前述的章節中提到，在不同“groupid”底下但是擁有相同的“ownerid”將被視為同一位使用者。所以“ownerid”在整個系統內最好是唯一值。

3.4 會議室主人會議狀態檔

MCU 在錄影檔目錄中維持兩組會議室主人會議 XML 狀態檔。第一組狀態檔採用的檔名格式為：

_status_ip_xxx.xml

‘xxx’ 是代表 MCU 伺服器的本機 IP 位址。MCU 伺服器的本機 IP 位址可以透過下列步驟來確定：

- 如果伺服器的授權檔中指定一組固定 IP 位址，它就會被指派為本機 IP 位址；
- 另外，如果在 config.ini 組態檔中有設定“listenip”，其值就會被指派為本機 IP 位址；
- 另外，如果伺服器有一張以上的網路卡，則第一張網路卡的 IP 位址就會被指派為本機 IP 位址。（即透過 gethostname 返回列表中的第一項）；
- 否則，將指派“127.0.0.1”為本機 IP 位址。

MCU 伺服器與其組態配置，必須明確地給予一個本機 IP 位址。這是非常重要的事，因為第三方應用程式需要透過這個資訊來找到特定 MCU 伺服器的狀態檔。

當使用 IPv6 IP 位址時，在 IPv6 IP 位址中所有的 ":" 將被置換成 ";" 做為檔名格式使用。例如，如果 MCU 的 IP 位址是 fe80::215:c5ff:fe0e:bef8，那麼相對應的檔案名稱就是 _status_ip_fe80;;215;c5ff;fe0e;bef8.xml。這種檔名格式會套用在所有使用 IP 位址的檔案上。

當在 config.ini 組態檔中設定 "jnj_ip"，且其值於前述步驟的本機 IP 位址不一樣時，將會使用配置的 "jnj_ip" 寫入第二個狀態檔。

一個 MCU 最多可以有三個狀態檔：

_status_ip_listenipv4.xml

_status_ip_listenipv6.xml

_status_ip_jnj_ip.xml

這三個檔案的內容是相同的。

下列顯示 XML 狀態檔範例：

```
<?xml version="1.0" encoding="utf-8" ?>
<ThirdParty utc="1477579954" last_update="Fri Apr 26 19:55:28">
  <info max_meeting="10" max_connection="20"
max_outside_connection="10" max_reserved_connection="5"
max_reserved_outside_connection="3" ip_range="209.101.242.123/16"
target_bitrate="24.0Kbps" test_wizard_link="wizard_mjnj"/>
  <directory>
    <meeting vid="0" session_index="0" name="John" userid="123"
meetingid="456" duration="30" elapsed_second1="303"
elapsed_second2="303" max_participant="6" owner_present="1"
have_guest="1" preparation_mode="1"/>
    <playback vid="0 session_index="2" name="player"
playerid="123" ip="192.168.1.10" elapsed_second="507"
shared_playback="1" end_reached="1" file_name="C:/Homemeeting/MMC
Server/data/recording/_user/2/169/_recording_192.168.1.21_2016_10_25
_21_07_001835.jnr" file_size="708125" file_duration="1298"
file_encrypted="0"/>
  </directory>
</ThirdParty>
```

這個檔案中 "last_update" 參數的時間狀態是這個 MCU 伺服器最後更新的時間。"utc" 元素與 "last_update" 相同，但是以機器友善的格式呈現。"info" 參數的值是指定這個 MCU 伺服器的屬性。第三方應用程式可以使用這些資訊來告知使用者有關伺服器的配置。當 MCU 伺服器停

止服務時，“info”元素中將沒有任何參數。

在“directory”元素底下的“meeting”（或“user”當主持人在會議中時）子元素是顯示正在進行會議的資訊。第三方應用程式可以使用這些資訊來顯示主持人的會議狀態給終端使用者。在“directory”元素底下的“playback”子元素是顯示正在播放錄影檔的資訊。

“meeting”元素中每個項目所代表的意義如下表所示：

項目	說明	預設值
vid	虛擬伺服器帳號	0
session_index	會議索引	-1
name	主持人名稱	空值
userid	主持人內部帳號	空值
meeting_id	會議帳號	空值
duration	預約型會議的持續時間（以秒為單位）	0
elapsed_second1	自會議起始以來的實際邏輯時間，以秒為單位。如果此會議是接續錄影檔，邏輯時間將涵蓋接續錄影檔的時間	0
elapsed_second2	自會議起始以來的實際真實時間，以秒為單位。如果此會議是接續錄影檔，真實時間將不會涵蓋接續錄影檔的時間	0
max_participant	曾經參加過會議的與會者人數	0
have_guest	會議中除了主持人之外還有其它與會者嗎？	0
preparation_mode	這是準備模式會議嗎？	0

“playback” 元素中每個項目所代表的意義如下表所示：

項目	說明	預設值
vid	虛擬伺服器帳號	0
session_index	會議索引	-1
name	播放者名稱	空值
playerid	播放者內部帳號	空值
ip	MCU 看見的播放者 IP 位址	空值
elapsed_second	自播放起始以來的實際經過時間，以秒為單位	0
shared_playback	這是播放共享錄影檔嗎？	0
end_reached	是否一直播放至錄影檔結尾？	0
file_name	錄影檔檔名	空值
file_size	檔案尺寸	0
file_duration	錄影檔持續時間，以秒為單位	0
file_encrypted	此檔案是否被加密？	0

由於啟用 HIT 的 MCU 支援即時型會議以及預約型會議，所以有可能一位使用者同時擁有多場線上會議。如果第三方應用程式要佈建讓訪客參加線上會議的介面，提供訪客與線上會議主持人相同的會議類型連結來參加會議是非常重要的。例如，如果使用者 “John” 在預約型會議中，要參加這個會議的訪客也應該使用預約型會議的請求。否則，可能會發生無法預期的錯誤。

如果第三方應用程式也支援 Messenger 介面，在這組狀態檔中將提供更多資訊。有關線上狀態的更多資訊將在後續章節說明。

第二組狀態檔採用的檔名格式為：

`_office_status_ip_xxx.xml`

此處的 ‘xxx’ 與之前章節的說明一樣。下列顯示 XML 狀態檔範例：

```
<?xml version="1.0" encoding="utf-8" ?>
<office_status>
  <user id="5" office_status="1" meeting_id="431"/>
  <user id="6" office_status="2" meeting_id="433"/>
</office_status>
```

這個檔案列出所有 MCU 伺服器上已開啟的會議室列表。

'office_status' 所代表的意義如下表所示：

0	會議室未開啟
1	會議室已開啟，但沒有其他訪客
2	會議室已開啟，且至少有一位以上的訪客
3	會議室已開啟，但使用者進入單人錄影模式

當系統中有多個 MCU 伺服器時，第三方應用程式必須使用這個 `_office_status_ip_xxx.xml` 狀態檔來確保在相同會議室中的所有與會者是被分配到同一個 MCU 伺服器。具體來說，當第三方應用程式準備產生一個即時型會議的 `jnj` 檔時，應該先掃描所有的 `_office_status_ip_xxx.xml` 狀態檔來檢查是否有任何主持人已開啟的會議室存在。如果在特定的 MCU 伺服器上有發現已開啟的會議室，則 `jnj` 檔應該分配到這個 MCU 伺服器。

3.5 錄影檔狀態檔

啟用 HIT 的 MCU 會幫每場會議保留一個單獨的 XML 檔。這個 XML 檔是跟錄影檔儲存在相同的目錄底下，且除了副檔名是從 `".jnr"` 變成 `".xml"` 之外，其餘檔名完全一樣。當使用者請求不錄影的 'meeting' 服務時，即使最終不會產生錄影檔，但還是會有 XML 檔產生。所以這個 XML 狀態檔可以用來檢查所有會議的歷程。

這個 XML 檔包含的資訊有會議持續時間、與會者以及會議標題等，這些可被第三方應用程式用來設計顯示更多有關錄影檔的詳細資訊。下列顯示 XML 狀態檔範例：

```
<?xml version="1.0" encoding="utf-8" ?>
<status>
  <starttime>20020426 15:51:00</starttime>
  <!-- duration is in seconds -->
  <duration>124</duration>
  <recording preparationmode="1" read="1086319845" shared="1"
view_count="3">true</recording>
  <title>Test Recording</title>
  <!-- how the meeting is terminated. 0: empty room; 1: owner's
request; 2: time is over; 3: server shutdown; 4: other reason -->
  <how>0</how>
  <participant userid="userid1" username="owner1" absent="1"/>
  <participant userid="userid2" username="guest" />
</status>
```

"title" 元素可以被視為是錄影檔的註釋。"title" 的內容是從使用者資訊中的 "mttingtitle"

元素所傳遞而來。

“how” 元素顯示會議室的終止方式。會議室的終止方式有：

- 沒有人在會議室，即所有與會者離開會議室；
- 會議室主人明確地請求終止會議室；
- 時間截止。預約型會議已經超過所預約的時間；
- 伺服器停止服務。MCU 伺服器停止服務，則所有進行中的會議室將被強迫終止。

這項元素可以用來避免會議室被多次地啟動。如果 “how” 元素為 “1” 或 “2”，會議室將不能再被啟動。如果為 “0”，情況將變得更加複雜。沒有人在會議室有可能是所有與會者有意離開會議室或是因為遭遇網路故障斷線所導致。在後者情況下，儘管伺服器已經將會議室終止，但與會者可能想要繼續未完成的會議。所以，如果相同的會議再次請求，會議伺服器還是要再次為使用者們開啟會議室。

“recording” 元素顯示是否會議進行錄影。如果 “recording” 元素擁有 “preparationmode” 參數，那麼相對應的錄影檔就是準備模式會議的結果。強烈建議，存取準備模式會議的錄影檔時要被限制僅能刪除，同時播放與下載則應該被禁用。如果 “recording” 元素擁有 “read” 參數，這是表示該錄影檔已被他自己的主人播放過，即該檔案不是 “new”。

自 MCU 3.9.0 版起，“recording” 元素有兩項新參數：(1) “shared” 是指錄影檔是否已共享，(2) “view_count” 是顯示錄影檔除了主人播放之外，已被其他人播放的次數。

3.6 與會者資訊檔

在會議進行時，MCU 會保留一個與會者資訊檔在錄影檔的 “participant” 子目錄中。這個檔案會列出在現行會議中所有上線的與會者。根據此檔案，第三方應用程式在必要時可以顯示相對應的與會者資訊。

這個檔的範例如下所示：

```
<?xml version="1.0" encoding="utf-8" ?>
<participant meetingid="241" guaranteed="0" utc="1477579954"
last_update="Sat Jun 05 00:02:28">
  <user name="a" userid="3" invited="1" ip="192.168.1.100"/>
</participant>
```

‘participant’ 是根標籤。此處的 ‘participant’ 有三項參數：‘meetingid’、‘guaranteed’ 以及 ‘last_update’。每個 “user” 子元素顯示在此會議中上線的與會者。

3.7 準備模式

預約會議的主人，即會議室主人，可以在預約型會議正式會議時間開始之前，使用準備模式會議來上傳簡報。只有會議室主人可以啟動預約型會議的準備模式。第三方應用程式可以產生分配“command”元素含“preparationmode”參數的 jnj 檔給會議室主人。會議室主人就可以利用這個 jnj 檔（例如“Prepare Meeting”）來啟動準備模式會議，接著準備上傳簡報。

準備模式會議的持續時間設定為一小時，且會自動延長時間。如果會議室主人終止準備模式會議，但想要準備更多簡報，可以再次請求準備模式會議，前一次準備模式會議的簡報將會自動轉移到這一次的準備模式會議中。如果有任何的準備模式會議啟動，而且簡報已經完成轉移，MCU 伺服器將會刪除前一次準備模式會議的錄影檔。所有準備模式會議的錄影檔會計算在會議室主人的磁碟用量中，而且準備模式會議的連線也會計算在連線人數中。

只有在準備模式會議的錄影檔中未被刪除的簡報，才會被轉移至一般會議中或是下一次的準備模式會議中；而其它所有資料，包括刪除的簡報、文字交談訊息、標記等將被捨棄。下列情境反應為準備模式會議的特性：

情境	反應
訪客想要啟動準備模式會議	[這是不可能發生的事情] 收到錯誤訊息 "Illegal operation"
當會議處於準備模式時，一位受到邀請的訪客想要參加會議	會議室主人會收到訪客請求參加會議的提示。如果接受請求，則會議將切換成一般模式；如果拒絕或忽略請求，訪客會收到 "會議室主人正在準備模式" 的提示。 如果會議室主人不在線上，訪客會進入會議室，則會議將切換成一般模式。
當會議處於準備模式時，一位未受到邀請的訪客想要參加會議	會議室主人會收到訪客請求參加會議的提示。如果接受請求，則會議將切換成一般模式；如果拒絕或忽略請求或會議室主人不在線上，訪客會收到如同一般會議模式一樣的錯誤訊息。
當會議處於準備模式時，會議室主人想要啟動準備模式會議	會議室主人繼續準備模式會議
當會議處於準備模式時，會議室主人想要參加會議	會議將切換成一般模式
當會議處於一般模式時，會議室主人想要啟動準備模式會議	會議室主人將收到 "會議已經開始，無法再進行準備模式" 的錯誤訊息
會議室主人想要啟動新的準備模式會議	如果一般模式會議已經終止，會議室主人會收到 "會議已結束" 的錯誤訊息；否則，會議室主人會啟動處於準備模式的準備模式會議

表二：準備模式會議屬性

當會議從準備模式切換成一般模式，所有上線的與會者會收到會議已切換成一般模式的提示。如果準備模式切換成一般模式，在準備模式期間所產生的文字交談訊息、標記都將被記錄下來。(請注意，如果將準備模式結束而非切換成一般模式，則這些資訊將被捨棄。)

準備模式會議的錄影檔在一般會議開始進行時就無法被刪除，且其中的簡報內容將被轉移至一般會議中。所以當 MCU 遇到服務中斷的情況時 (例如當機、停電等)，在一般會議重新啟動之

後，準備模式的簡報依然可用。

強烈建議，存取準備模式會議的錄影檔時要被限制僅能刪除，而播放與下載則應該被禁用。也建議在查詢會議進行搜尋時，不要列出準備模式會議。

當會議室主人正在準備模式會議時，會議室主人的線上狀態會包含“preparationmode”參數。第三方應用程式可以自行決定要顯示會議室主人正在會議或不在會議。

3.8 伺服器備援

要持續提供會議服務最好是能夠設置一個或多個備援會議伺服器，以避免因為主要的 MCU 關機或維護而停止服務。

要建立備援 MCU 伺服器有幾項要求：

- 這些 MCU 伺服器的錄影檔目錄必須是在相同的路徑。這表示必須存在檔案共用系統；
- 所有的 MCU 伺服器務必使用**相同金鑰對**。這是為了確保所有的 MCU 伺服器可以解開由第三方應用程式所產生的 jnj 檔；
- 第三方應用程式要知道所有的 MCU 伺服器。

當使用備援伺服器時，第三方應用程式必須在產生的 jnj 檔中放入更多項目：

```
backupip=backupip1,backupip2,backupip3
```

如果有多個備援伺服器時，它們的 IP 位址在 jnj 檔中應該以“,”逗點分隔。截至目前為止，最多只支援 9 個備援伺服器。如果 JoinNet 無法連接到主要伺服器，它將會按照在 jnj 檔中的順序來嘗試連接備援伺服器。

在 jnj 檔中還有另外一個補充的項目：“portm2”。“portm2”是指定伺服器正在聆聽的第二個通訊埠。

JoinNet 會依照下列順序來嘗試所有伺服器/通訊埠組合：

```
primary_server/portm→primary_server/portm2→backup_server_1/portm→  
backup_server_2/portm2→…→last_backup_server/portm→last_backup_server/portm2
```

請注意，這些嘗試連線是平行的，在兩個連續的伺服器/通訊埠連線之間的延遲時間為 500 毫秒。所有組合中的第一個連線被採用之後，其它正在嘗試的連線將被捨棄。

請注意，當使用者指定使用代理伺服器且 portm2 為 '443' 時，portm 與 portm2 將會被交換（即當使用代理伺服器時，會先嘗試 443 通訊埠）。

3.9 伺服器叢集與負載平衡

比前面所述的伺服器備援更進階的應用就是同時將所有的會議伺服器連接在一起提供服務。這麼做，所有的會議伺服器連接在一起可以被看作是一個伺服器叢集，此外在所有的會議伺服器之間也可以使用負載平衡。

伺服器叢集的需求與伺服器備援相同。然而，要啟動負載平衡，在每個產生的 `jnj` 檔中主伺服器與備援伺服器的順序必須是動態的。

我們使用一個簡單的範例來說明如何設定伺服器叢集。假設有三個 MCU 伺服器分別在 192.168.1.1、192.168.1.2 以及 192.168.1.3。錄影檔目錄是位在一个單獨的儲存伺服器 192.168.1.10。所有的 MCU 伺服器必須設定

```
recording=\\192.168.1.10\recording
```

在它們的組態檔中 (`configm.ini`)。

負載平衡是透過幫每個不同會議的 `jnj` 檔分配不同的主伺服器與備援伺服器順序來實現配置。幫同一個會議 (即相同的 "meetingid") 的所有 `jnj` 檔分配相同的伺服器順序是非常重要的；否則，同一個會議的與會者可能會連接到不同的伺服器，因此就無法參加同一個會議。

有兩種方式可以實現負載平衡，一種是統計的負載平衡，另外一種是限制的負載平衡。

在統計的負載平衡演算法中，每個會議的 `jnj` 檔中的伺服器順序是隨機分配的。三個伺服器就有六種不同的伺服器順序可以選擇。例如其中一種順序是：

```
ip=192.168.1.2
```

```
backupip=192.168.1.3,192.168.1.1
```

在這種統計的演算法之下，第三方應用程式並不需要知道會議伺服器的狀態。負載平衡純粹由統計來實現。請注意，如果每個伺服器的容量 (容量資訊取決於伺服器的授權) 不同，建議採用容量加權演算法。

在限制的負載平衡演算法之下，第三方應用程式必須時時刻刻知道每個伺服器的負載。每當產生一個新的會議，最好的負載候選伺服器是經過現行負載狀態以及在新會議中已知的與會者資訊計算之後才決定的。第三方應用程式可以自己決定使用任何一種演算法來實現負載平衡，本文將不會提供任何範例。

每個伺服器的負載與啟動狀態可以透過讀取在錄影檔目錄中的狀態檔來進行檢索。每個伺服器在錄影檔目錄中會維護兩個狀態檔，一個是啟動狀態檔，另一個是負載狀態檔。取 192.168.1.1

伺服器為例，其啟動狀態檔是

```
_cluster_ip_192.168.1.1.txt
```

負載狀態檔是

```
_cluster_connection_192.168.1.1_.txt
```

請注意，在負載狀態檔 IP 位址的後方有個 “_” 符號。當相對應的伺服器狀態為啟動時，啟動狀態檔的內容為 “1”；當它停止時內容為 “0”。

負載狀態檔的內容為一行由空格分隔的十四組（自 3.23.8 版起新增最後四組數字）數字：

“現行會議室數” “會議室數上限” “現行總連線數” “總連線數上限”

“現行外部總連線數” “外部總連線數上限” “現行預約型會議總連線數”

“預約型會議連線數上限” “現行預約型會議外部總連線數” “預約型會議外部總連線數上限”

“現行總播放連線數” “總播放連線數上限” “現行外部總播放連線數” “外部總播放連線數上限”

例如，一臺伺服器具有

```
8 60 23 300 19 100 5 10 3 8 7 20 6 15
```

在其負載狀態檔中，這是表示

- ✓ 目前伺服器擁有 8 間會議室、總連線數 23 個、其中有 19 個屬於外部總連線數；
- ✓ 這個伺服器最多可以支援的內容是會議室數上限 60 間、總連線數上限 300 個、外部總連線數上限 100 個；
- ✓ 目前伺服器擁有預約型會議總連線數 5 個、其中有 3 個屬於預約型會議外部總連線數；
- ✓ 這個伺服器最多可以支援的內容是預約型會議總連線數上限 10 個、預約型會議外部總連線數上限 8 個；
- ✓ 目前伺服器擁有總播放連線數 7 個、其中有 6 個屬於外部播放連線數；
- ✓ 這個伺服器最多可以支援的內容是總播放連線數上限 20 個、外部總播放連線數上限 15 個。

再舉另外一個例子，一臺伺服器具有（播放連線數被視為連線數的一部分）

```
8 60 23 300 19 100 5 10 3 8 7 0 6 0
```

在其負載狀態檔中，這是表示

- ✓ 目前伺服器擁有 8 間會議室、總連線數 23 個、其中有 19 個屬於外部總連線數；

- ✓ 這個伺服器最多可以支援的內容是會議室數上限 60 間、總連線數上限 300 個、外部總連線數上限 100 個；
- ✓ 目前伺服器擁有預約型會議總連線數 5 個、其中有 3 個屬於預約型會議外部總連線數；
- ✓ 這個伺服器最多可以支援的內容是預約型會議總連線數上限 10 個、預約型會議外部總連線數上限 8 個；
- ✓ 總連線數 23 個、其中有 7 個屬於總播放連線數、其它 16 (23-7) 個屬於總會議連線數；
- ✓ 外部總連線數 19 個、其中有 6 個屬於外部總播放連線數、其它 13 (19-6) 個屬於總外部會議連線數。

播放連線數獨立計算或是合併計算是由授權檔的 `<max_playback_connection>` 與 `<max_playback_outconnection>` 項目所控制。

當多個會議伺服器組成一個伺服器叢集時，會議伺服器將不會 SPOF (單點故障，全部故障)。然而，有另外兩種 SPOF 情況：就是第三方應用程式以及儲存伺服器。當第三方應用程式停止服務時，終端使用者將無法取得有效的 jnj 進而啟動 JoinNet。當儲存伺服器停止服務時，終端使用者將無法進行播放以及下載錄影檔，且會議將不會進行錄影 (會議功能依舊可以運作，只有錄影功能失效)。

3.10 密碼驗證

預設，HIT 使用 jnj 的有效期來驗證終端使用者。然而，HIT 也支援 MCU 伺服器與終端使用者之間的密碼驗證。

由於第三方應用程式擁有終端使用者的密碼資訊，所以它可以將使用者密碼以 base64 加密進行安全雜湊演算法後指派給 jnj 檔中使用者資訊的 "password" 元素。當終端使用者連接到會議伺服器，會議伺服器就會要求終端使用者輸入密碼。在終端使用者輸入密碼之後，其所輸入的安全雜湊演算法密碼會經過計算並透過 JoinNet 與會議伺服器之間的安全通道傳遞給會議伺服器。會議伺服器可以經由比對來自終端使用者的雜湊與第三方應用程式的雜湊來驗證終端使用者身分。

在使用者資訊中的 "timestamp" 元素是用來指示密碼的到期時間。在這個時間戳記之後密碼將被停用。基於安全考量，到期時間不宜過久。如果攻擊者截獲 jnj 檔，並且從其它途徑得知密碼，他將可以偽裝成真正的使用者。一個合適的到期時間 (例如，一天後) 可以避免這類的攻擊發

生。

如果 JoinNet 支援儲存成功登入的密碼，終端使用者將不會每次都收到輸入密碼的提示。

由於密碼驗證會提示終端使用者輸入他們的密碼，這樣的設計對使用者來說可能不太方便，因此建議在機密會議時才使用。

下列有兩個密碼驗證用法的範例。

範例一：(在此範例中會使用到使用者的帳號密碼) John 邀請 Steven 參加會議，該會議為機密會議，所以他選擇在第三方應用程式設定使用密碼驗證。會議邀請函經由電子郵件寄送給 John 以及 Steven。寄給 John 的會議 jnj 檔包含 John 密碼的雜湊，同時 Steven 的會議 jnj 檔一樣也包含 Steven 密碼的雜湊。這些 jnj 檔可以附加在電子郵件中；所以後來當 John 與 Steven 點選 jnj 檔來啟動 JoinNet 進入會議室時，就會收到輸入密碼的提示。密碼的到期時間 ('timestamp' 參數) 建議可以設置與會議結束時間一致。

範例二：(使用者可以為機密會議設置特殊入口代碼) John 邀請 Steven 與 Mike 參加會議，該會議為機密會議。John 選擇在第三方應用程式介面設定指派一組會議密碼。會議邀請函經由電子郵件寄送給 John、Steven 以及 Mike。這三個人的會議 jnj 檔都會包含由 John 所指派密碼的雜湊。這些 jnj 檔可以附加在電子郵件中，所以後來當 John、Steven 以及 Mike 點選 jnj 檔來啟動 JoinNet 時，就可以輸入所需的密碼。密碼的到期時間 ('timestamp' 參數) 建議可以設置與會議結束時間一致。

第三方應用程式可以自己決定使用任何一種範例來使用密碼驗證，或是使用其它更安全運用的方式。這項密碼驗證是非必要的功能。如果以方便使用為前提，第三方應用程式可以只使用預設的驗證方式，也就是取決於 jnj 檔的有效期。

請注意，如果 jnj 檔是透過安全通道遞送給一般使用者，例如 HTTPS，則預設的驗證方式也被認為是安全的，可以應用在機密會議上。

3.11 第三方應用程式規則

對 HIT 系統而言，第三方應用程式負責下列工作：

- 管理與驗證使用者身分。(所有使用者可以透過使用資料庫、簡單的純文字檔或其它技術來進行管理。建議使用如安全的 HTTPS 網頁介面來驗證；)
- 為使用者產生請求服務的 jnj 檔來連結 MCU；
- 呈現會議室主人上線與離線狀態給終端使用者；

- 從網頁列出主人的錄影檔，且支援下載與刪除錄影檔。(請注意，當刪除錄影檔時，相對應的 XML 狀態檔也會被第三方應用程式刪除；)
- 如果主人想要公佈他的錄影檔給所有人，要產生一個可以適合給任何人使用的 jnj 檔。

3.12 錄影檔保管

自 MCU 3.10 版起，支援錄影檔保管功能，也就是說，一旦某個錄影檔被 MCU 保管，即使在使用者刪除錄影檔之後，系統管理者還是可以回復該錄影檔。在 config.ini 檔中有兩項設定與這個功能有關。

第一個設定是 "archive_recording_file"。系統管理者可以設定這項參數的值不為零，就可以啟用保管功能。當某個錄影檔被其主人 "刪除"，錄影檔並不會從 MCU 刪除，而是將其重新命名成不同的檔案名稱。如果 "archive_recording_file" 的數值設定成正數，這是表示保管被刪除錄影檔的天數，等到配置的時間到期時，MCU 會自動將保管的錄影檔移除。然而，如果 "archive_recording_file" 的數值設定成任何負數，MCU 將不會自動移除保管的錄影檔。

第二個設定是 "disable_jnr_password"。當設定這項參數的值不為零時，不管是經由會議或是變更 JNR 密碼，MCU 都不會產生密碼保護的錄影檔。然而，使用者還是可以上傳任何密碼保護的錄影檔。

在錄影檔被保管之前，他的主人必須事先收到警告。如果沒有收到警告，錄影檔不該被保管。根據這個規則，當系統啟用保管功能時，現有的錄影檔將不會被保管。只有新產生的錄影檔會進行保管。當會議產生錄影檔時，在會議中的所有 JoinNet 使用者都會收到錄影檔將被保管的提示。在使用者上傳錄影檔之前，使用者也會收到錄影檔將被保管的提示。如同在 3.5 章節中所述的，在相對應的錄影狀態檔中，這些錄影檔會使用 <status> 根元素的 "to_archive" 參數進行標籤。例如：

```
<?xml version="1.0" encoding="utf-8" ?>
<status to_archive="1">
  <starttime>20020426 15:51:00</starttime>
  <!-- duration is in seconds -->
  <duration>124</duration>
  <recording preparationmode="1" read="1086319845" shared="1"
view_count="3">true</recording>
  <title>Test Recording</title>
  <!-- how the meeting is terminated. 0: empty room; 1: owner's
request; 2: time is over; 3: server shutdown; 4: other reason -->
  <how>0</how>
  <participant userid="userid1" username="owner1" absent="1"/>
  <participant userid="userid2" username="guest" />
</status>
```


當啟用保管時，無論使用者是否請求錄影，MCU 都會強制進行錄影。基於這個原因，最好第三方應用程式也強制錄影。此外，當啟用保管時，JoinNet 使用者在會議進行中將無法暫停錄影。

當使用者刪除某個在相對應的狀態檔中擁有 "to_archive" 參數的錄影檔時，檔案將被重新命名，而不是被刪除。例如，一個錄影檔的檔案名稱原本是 "test.jnr"，將被重新命名成 "test.archive_jnr_xxxx"，這裡的 "xxxx" 是刪除的時間。"xxxx" 是使用 C 語言 time () 函式庫所取得的本機時間。如果使用者透過第三方應用程式介面想要刪除即將被保管的錄影檔，第三方應用程式應該遵循此規則將錄影檔重新命名。

在啟用保管狀態下，當使用者想要透過第三方應用程式介面上傳錄影檔，且相對應的狀態檔擁有 "to_archive" 參數時，第三方應用程式會被催促要提示使用者。

3.13 一次性 MCU Relay

自 JoinNet 4.0 版起，JoinNet 支援在 jnj 檔中的 "use_relay" 項目。這是讓 JoinNet 使用指定的一次性 MCU Relay 來連結 MCU。MCU Relay 的資訊將不會被儲存下來，且不會影響其它會議。

"use_relay" 的格式是

```
use_relay=mcu_relay,port
```

上述 mcu_relay 的值可以是網域名稱也可以是 IP 位址。通訊埠資訊是必要的，而逗號是用來分隔 Relay 與通訊埠。

肆、JoinNet Messenger

HIT 也支援 JoinNet Messenger 系統，其中使用者的出席資訊可以進行管理。要支援 Messenger 系統，MCU (3.0 版以上) 必須成為 Messenger 伺服器，來維護使用者的 Messenger 連線以及傳送會議請求給第三方應用程式上的 Messenger 使用者。在 MCU 與第三方應用程式之間的請求與回應均採用與傳遞驗證一樣的加密方式。

4.1 MCU 的組態配置

在 MCU 的 configm.ini 檔中有兩項設定跟 MCU 與第三方應用程式之間的安全互動有關。

```
messenger_web_url = http://mmc.homemeeting.com:8080/~webservice.php
```

```
messenger_dh_key = key_web_messenger_a.b.c.d_2333.x509
```

messenger_web_url 是指定請求要被傳送的網址。而 messenger_dh_key 則是指定第三方應用程式用來加密請求與解密回應的公鑰。

自 3.3.0 版起，在 configm.ini 檔中新增一項設定 'messenger_secure_channel'。當這個項目設定為 0 時，除了 'user authentication' 請求是永遠加密之外，來自 MCU 傳送給第三方應用程式的請求是加密的。第三方應用程式必須檢查來源 IP 位址或是其它資訊以確保請求是來自一個可靠的 MCU 伺服器。

4.2 獲取 MCU 的 IP 與 Port

第三方應用程式的網頁必須在一個請求

```
http://thirdparty.homepage.com/getmcuinfo/?userid=abcde
```

返回 MCU 的 IP 位址與通訊埠資訊，否則，此第三方應用程式將會出現沒有 Messenger 服務。

MCU 的資訊寫入在 XML 格式中，例如，

```
<?xml version="1.0" encoding="utf-8" ?>
<mcuinfo>
  <ip>192.168.1.192</ip>
  <backupip>192.168.1.193</backupip>
  <port>443</port>
  <port2>2333</port2>
</mcuinfo>
```

4.3 MCU 與第三方應用程式之間的互動 (由 MCU 發起的互動)

MCU 與第三方應用程式之間互動的傳遞驗證演算法與用來產生 jnj 檔的類似。

來自 MCU 的請求寫入在 XML 格式中，這裡有個初步的範例：

```
<?xml version="1.0" encoding="utf-8" ?>
<msg_r_request>
  <disk_quota_alert internal_id="a"/>
</msg_r_request>
```

這個 XML 檔會經過加密、簽名並串連一個 http 請求來傳送

```
http://messenger.web.url?param=key_mcu_192.168.1.63_0000|cIyqQrp7
VQdnvb6EN5pUvA3v90+43S59SbcRztr0EAg=|OukU+lvEeDaXyiLi7T7KJ+Iud4sd
vudnFDqN4/dzMDQ4y21pZtf38ds3yssSyODuCXlK+jbfmutqtflP9+HLHZmPe0Y7U
QAZdJUyo+twmqpwly8i6bZAs5Z/N800dY7gN2YTGKpM4C2XbQOTooUQ1W9Tq5vMa
nz82Uin65Nwjly2X6aD1m1/4tXrRXpt84NxAfe0ZxAyOa4sohnV8u+KLbP3baQzry
HQYU/UE7VwFIkfz2dD0NjLj2B6uL32ZrOIIMShGEyk/GMBjTEp9KlCYsYn8ne564S
wf6bwxK5K5zCkeJkyrn4WnchtJjoGQhP+2sPJY90cSn1MYFRrDH6FQ3tw/hvWZftl
WMjeBNEZJX9wDgF6TsBG5AHaORkbGa3t98=|n4xpLtt0GbkhsAr8dLA7iXK43gg=
```

來自第三方應用程式的回應會被封裝在 http 的回應裡，就類似在 jnj 檔中的 userinfo2 內容。

當 MCU 與第三方應用程式之間的路徑被認定是安全時，請求可以傳送不經過加密的純文字。這是經由在 config.ini 檔中的 "messenger_secure_channel" 設定所控制。當 "messenger_secure_channel" 設定為 0，MCU 將使用純文字與第三方應用程式之間進行通訊。在這種情況下，網頁必須檢查請求的來源 IP 位址以確認請求來自其授權的 MCU 伺服器。否則，攻擊者可以利用未經授權的請求來獲取其他人的檔案。第三方應用程式可以在 mmc.ini 檔中使用 "authorizedIp" 設定來指定附加授權的 IP 位址。

4.4 請求列表

在本章節中將介紹所有從 MCU 傳送給第三方應用程式的請求與其相對應的回應。如果第三方應用程式的金鑰出現問題（不管是無法解密訊息，或是很難加密回應），它應該返回一個非 200 回應且帶有詳細錯誤訊息內容的 HTTP 網頁。請注意，包含在回應的錯誤代碼必須介於 20000 到 29999 之間。如果已包括詳細的錯誤內容，錯誤代碼就為非必要。在回應中的 "public_status" 可以為 "1" 或 "0"，其中 "1" 是指使用者為公開的，也就是說，所有人可以看見這位使用者。在回應中的 "sig" 表示簽名（如雜湊的內容）的內容。如果在請求中的 "sig" 數值與新回應的簽名相符，則可以假定 MCU 已經具有相同的內容，這樣的回應可以略過內容，並將 "no_change" 設

定為 1。

幾個獨立的請求可以被整合在一個批次的請求。每個獨立的請求應具備 'seq' 參數來識別自己。批次請求的回應應該保持其先後順序。對於批次請求的部份回應是被允許的。

1) user authentication (MCU 3.0 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <user_auth seq="0" userid="a" passwd="12345"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_response> <user_auth seq="0" passed="1" errorcode="20001" internal_id="a" name="a" public_status="1"> detailed error is written here if not passed </user_auth> </msgr_reponse></pre>

請注意，'seq' 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 'seq'，這個參數在回應中必須原封不動的重複呈現。

請注意，在請求中的 'userid' 是用來做為登入帳號使用的使用者帳號。在回應中的 'internal_id' 是由第三方應用程式所使用的內部使用者帳號，是做為代表使用者的唯一值。這兩個 'id' 可能不相同，也可能相同，這將視第三方應用程式個別的施行方式而定。"public_status" 顯示使用者是否為公開可見的：其值為 "1" 表示使用者可以被所有人看見；當其值為 "0" 表示使用者僅能被特定的使用者看見。

請注意，如果認證成功，MCU 會暫存密碼與返回的內容達一小時。

2) user profile (MCU 3.0 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <user_profile seq="0" internal_id="a"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_response> <user_profile seq="0" passed="1" errorcode="20001" name="a" public_status="1"> detailed error is written here if not passed </user_profile> </msgr_reponse></pre>

請注意，'seq' 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求 HomeMeeting 技術文件：HomeMeeting 整合工具 33

中存在 'seq'，這個參數在回應中必須原封不動的重複呈現。

3) query account info (MCU 3.0 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <query_info seq="0" internal_id="a"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_response> <query_info seq="0" passed="1" errorcode="20001"> detailed error is written here if not passed. <name>name</name> <email>email</email> <maxoutconnection>100</maxoutconnection> <diskquota used="27">1000</diskquota> <message new="3">7</message> <meeting new="11">24</meeting> </query_info> </msgr_reponse></pre>

請注意，'seq' 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 'seq'，這個參數在回應中必須原封不動的重複呈現。

對第三方應用程式來說，根據相對應錄影檔的 xml 檔，一個錄影檔可以被確定為留言或是會議錄影檔。如果在錄影檔中會議室主人（在 xml 檔中第一位與會者）缺席（與會者標籤設定 'absent' 參數），這會被認定是留言；除此之外則被認定是會議。

對第三方應用程式來說，經由檢查相對應錄影檔的 xml 檔，留言/會議錄影檔可以被確定為未讀取或已讀取。如果 'recording' 標籤設定 'read' 參數，則被認定是已讀取；除此之外則被認定是未讀取。"<diskquota>" 的單位為 MB。

4) launch office (MCU 3.0 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <launch_office internal_id="a" meeting_id="meetingid"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_response> <launch_office passed="1" errorcode="20001"> detailed error is written here if not passed. <jnj>base64 encoded jnj file content</jnj> </launch_office> </msgr_reponse></pre>

請注意：如果會議室主人尚未開啟會議室，則 meeting_id 有可能是空值。

5) owner_visit_owner (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <owner_visit_owner visitor_internal_id="visitor"
  host_internal_id="host" meeting_id="meeting_id"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <owner_visit_owner passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </owner_visit_owner>
</msgr_reponse>
```

6) guest_visit_owner (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <guest_visit_owner visitor_name="visitor" host_internal_id="host"
  meeting_id="meeting_id"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <guest_visit_owner passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </guest_visit_owner>
</msgr_reponse>
```

7) owner_leave_message (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <owner_leave_message visitor_internal_id="visitor"
  host_internal_id="host"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <owner_leave_message passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </owner_leave_message>
</msgr_reponse>
```

8) guest_leave_message (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <guest_leave_message visitor_name="visitor"
host_internal_id="host"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <guest_leave_message passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </guest_leave_message>
</msgr_reponse>
```

9) owner_invite_guest (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <owner_invite_guest visitor_name="visitor" host_internal_id="host"
meeting_id="meeting_id"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <owner_invite_guest passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </owner_invite_guest>
</msgr_reponse>
```

請注意，對 owner_invite_guest 的 jnj 來說，其 'invited' 值應該設定為 1。此外，由於這是**明確**的邀請，即使主人設置其自己僅能被聯絡人看見，這個請求還是必須產生一個 jnj 檔。

10) owner_invite_owner (MCU 3.0 版以上)

request.xml :

```
<msgr_request>
  <owner_invite_guest visitor_id="visitor" host_internal_id="host"
meeting_id="meeting_id"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <owner_invite_owner passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </owner_invite_owner>
</msgr_reponse>
```

11) allow list (MCU 3.3 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msggr_request>
  <allow_list seq="0" internal_id="a" sig="abcdefg"/>
</msggr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msggr_response>
  <allow_list seq="0" passed="1" errorcode="20001" no_change="0"
sig="abcdefg" public_status="0">
  <i>a</i>
  <i>b</i>
  <i>c</i>
</allow_list>
</msggr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

這個請求查詢可被看見的使用者所提供的 `internal_id` 列表。在這個請求中，`sig` 參數是最後回應所接收到的簽名。如果自前次請求以來沒有任何改變，網頁可以將 `no_change` 設置為 1，並且無須提供使用者列表。請注意，`public_status` 必須被正確地配置。如果使用者的 `public_status` 變動或是他的聯絡人列表變動，`no_change` 應該設置為 1。`<i>` 子元素是顯示在請求中所提供允許被看見的所有使用者的 `internal_id`。

12) meeting start notification (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msggr_request>
  <meeting_start_notification seq="0" internal_id="a" meeting_id="m"
mcu_ip="192.168.1.8" jnj_ip="mmc.homemeeting.com" utc="123456789"/>
</msggr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msggr_response>
  <meeting_start_notification seq="0" passed="1" errorcode="20001"/>
</msggr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，`mcu_ip` 是指定十進制格式的 IP 位址，而 `jnj_ip` 是指定在 `config.ini` 檔中所配置的內容。這個通知將告訴第三方應用程式某個會議識別碼會議確切的 MCU 位置，以便引導其他與會者至相同的 MCU。此處，`utc` 是指定由 MCU 所傳送請求的時間。

13) meeting end notification (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <meeting_end_notification seq="0" internal_id="a" meeting_id="m"
file_type="message" visitor_name="v" utc="123456789"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <meeting_end_notification seq="0" passed="1" errorcode="20001"/>
</msgr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，當錄影檔類型為 `message` (另一種是 `meeting`) 時，`vistor_name` 參數才有意義。再者，`utc` 是指定由 MCU 所傳送請求的時間。

14) playback start notification (MCU 3.3 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <playback_start_notification seq="0" internal_id="a" meeting_id="m"
file_name="" utc="123456789" player_id="a" player_name="b"
ip="192.168.1.1"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <playback_start_notification seq="0" passed="1" errorcode="20001"/>
</msgr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，`file_name` 不能有副檔名 (如 `userinfo2` 所規定。)

15) playback end notification (MCU 3.3 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <playback_end_notification seq="0" internal_id="a" meeting_id="m"
file_name="" utc="123456789" player_id="a" player_name="b"
duration="222" data_received="1" ip="192.168.1.1"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <playback_end_notification seq="0" passed="1" errorcode="20001"/>
```

```
</msgr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，`file_name` 不能有副檔名（如 `userinfo2` 所規定。）此處，`utc` 是指定由 MCU 所傳送請求的時間；`duration` 的單位是毫秒，而 `data_received` 是顯示在播放錄影檔時是否接收任何來自伺服器的資料。

16) download start notification (MCU 3.3 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <download_start_notification seq="0" internal_id="a" meeting_id="m"
file_name="" utc="123456789" player_id="a" player_name="b"
ip="192.168.1.1"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <download_start_notification seq="0" passed="1" errorcode="20001"/>
</msgr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，`file_name` 不能有副檔名（如 `userinfo2` 所規定。）

17) download end notification (MCU 3.3 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <download_end_notification seq="0" internal_id="a" meeting_id="m"
file_name="" utc="123456789" player_id="a" player_name="b"
duration="222" completed="1" ip="192.168.1.1"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <download_end_notification seq="0" passed="1" errorcode="20001"/>
</msgr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，`file_name` 不能有副檔名（如 `userinfo2` 所規定。）此處，`utc` 是指定由 MCU 所傳送請求的時間，`duration` 的單位是毫秒，而 `completed` 是顯示該錄影檔是否已被使用者

完全下載。

18) disk quota alert (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <disk_quota_alert seq="0" internal_id="a"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <disk_quota_alert seq="0" passed="1" errorcode="20001"/>
</msgr_reponse>
```

19) query web url (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <query_url internal_id="a" type="homepage"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <query_url passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <url>base64 encoded url </url>
  </query_url>
</msgr_reponse>
```

請注意，type 有可能是 "homepage"、"checkmessage" 或 "editprofile" 其中一種。當 type 為 "checkmessage" 且 internal_id 為 "blank" (即，這是個訪客) 時，第三方應用程式可以自行決定是回應一個空白網頁或是指向一個公開的錄影檔網址。Type 為 "checkmessage" 以及 "editprofile" 返回的網址應該是短暫的，也就是說，它應該在一段時間之後失效，例如五分鐘後。

20) public user list (MCU 3.0 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <public_user_list internal_id="a" hint="test" />
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <public_user_list passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <user internal_id="1" userid="a@a.com" username="a"></user>
```

```
<user internal_id="2" userid="b@b.com" username="b"></user>
</public_user>
</msgr_reponse>
```

請注意，返回的使用者其個人設定均為可被所有人看見。而 `userid` 為登入的帳號，其值可能與 `internal_id` 不同。

21) command list (MCU 3.3 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <command_list seq="0"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <command_list seq="0" passed="1" errorcode="20001">
    all supported commands are put here, in lower case, separated
    by comma.
  </command_list>
</msgr_reponse>
```

請注意，`'seq'` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `'seq'`，這個參數在回應中必須原封不動的重複呈現。

MCU 伺服器使用這個請求來找出網頁支援的指令。當第三方應用程式啟動時，它應該通知 MCU 來請求指令列表。這個通知是經由第三方應用程式與 MCU 之間的互動來進行傳送（請見 4.8 章節）。請注意，逗號是唯一用來分隔指令列表的有效符號。

22) import contact (MCU 3.6 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <import_contact internal_id="abc"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <import_contact internal_id="abc" passed="1" errorcode="20001">
    <contactgroup groupname="c">
      <contact userid="a" username="aa"/>
      <contact userid="b" username="bb"/>
    </contactgroup>
  </import_contact/>
</msgr_reponse>
```

MCU 使用這個請求來要求某位特定使用者的聯絡人及群組資訊。請注意，在 `<contact>` 中這裡的 `'userid'` 標籤是指 `'internal_id'`。

23) export contact (MCU 3.6 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <export_contact internal_id="abc">
    <contactgroup groupname="c">
      <contact userid="a"/>
      <contact userid="b"/>
    </contactgroup>
  </export_contact>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <export_contact internal_id="abc" passed="1" errorcode="20001">
  </export_contact>
</msgr_reponse>
```

MCU 使用這個請求來匯出聯絡人及群組資訊給第三方應用程式。請注意，在 `<contact>` 中這裡的 `'userid'` 標籤是指 `'internal_id'`。

24) query event (MCU 3.6 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <query_event seq="0" sig="abc"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <query_event seq="0" passed="1" errorcode="20001" no_change="0"
  sig="abcdefg">
    <event meeting_id="" title="" owner_id="" owner_name=""
    public_status="" start_time="" duration="" advance="">
      <user>abc</user>
      <user>def</user>
      <user>fff</user>
    </event>
  </query_event>
</msgr_reponse>
```

請注意，`'seq'` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `'seq'`，這個參數在回應中必須原封不動的重複呈現。

這個請求是查詢預約型會議/活動的列表。在這個請求中，`'sig'` 參數是最後回應所接收到的簽名。如果自前次請求以來沒有任何改變，網頁可以將 `'no_change'` 設置為 1，並且無須提供活動列表。`'public_status'` 參數是指定是否所有人可以看見這個活動。當 `'public_status'` 設定為 `'1'`

時，所有註冊的使用者可以看見這個活動。當 `public_status` 設定為 `2` 時，所有人（包括訪客）均能看見這個活動。`start_time` 參數是指定預約型活動的開始時間（由 C 語言的 `time()` 函數所返回的值）。`duration` 參數是指定活動的持續時間（以秒為單位）。當活動持續時間被自動地延長時，第三方應用程式可以設定 `duration` 為 0。`advance` 參數是指定會員可在預約時間開始多久前（以秒為單位）參加活動。`<user>` 項目列出這個活動所有受邀請的註冊會員。`<user>` 的內容是顯示使用者的 `internal_id`。

25) load client info (MCU 3.8 到 MCU 3.14 版)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <load_ci internal_id="abc"/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <load_ci internal_id="abc" passed="1" errorcode="20001">
    <ci office_name="Test">
      <contactgroup groupname="c">
        <contact userid="a" username="aa"/>
        <contact userid="b" username="bb"/>
      </contactgroup>
    </ci>
  </load_ci/>
</msgr_reponse>
```

MCU 使用這個請求從第三方應用程式來載入指定使用者的使用者資訊。請注意，第三方應用程式已經事先透過一個來自 MCU 的 `save_ci` 請求（請見下一段說明）來儲存使用者資訊。如果這類的資訊無法使用，第三方應用程式應該返回一個錯誤。

自 MCU 3.15 版起，MCU 直接在錄影檔目錄中儲存使用者資訊。例如，使用者 `userid` 的使用者資訊是儲存在 `~recording/_user/userid/_client_info_userid.dat` 檔中。

26) save client information (MCU 3.8 到 MCU 3.14 版)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <save_ci internal_id="abc">
    <ci office_name="Test">
      <contactgroup groupname="c">
        <contact userid="a" username="aa"/>
        <contact userid="b" username="bb"/>
      </contactgroup>
    </ci>
  </save_ci>
</msgr_request>
```

```

</save_ci>
</msgr_request>
response.xml :
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <save_ci internal_id="abc" passed="1" errorcode="20001">
  </save_ci>
</msgr_reponse>

```

MCU 使用這個請求讓第三方應用或是儲存指定使用者的使用者資訊。請注意，第三方應用程式無須了解使用者資訊的語法。內容可以被完全原封不動地儲存。

自 MCU 3.15 版起，MCU 直接在錄影檔目錄中儲存使用者資訊。例如，使用者 "userid" 的使用者資訊是儲存在 ~recording/_user/userid/_client_info_userid.dat 檔中。

27) query_publish_file (MCU 3.9 版以上)

```

request.xml :
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <query_publish_file seq="0" sig="abc"/>
</msgr_request>
response.xml :
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <query_publish_file seq="0" passed="1" errorcode="20001"
no_change="0" sig="abcdefg">
  <publish_file blackbox_data="" title="" owner_name=""
  public_status="" duration="">
    <user>abc</user>
    <user>def</user>
    <user>fff</user>
  </publish_file>
</query_publish_file>
</msgr_reponse>

```

請注意，'seq' 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 'seq'，這個參數在回應中必須原封不動的重複呈現。

這個請求是查詢公告錄影檔的列表。在這個請求中，'sig' 參數是最後回應所接收到的簽名。如果自前次請求以來沒有任何改變，網頁可以將 'no_change' 設置為 1，並且無須提供檔案列表。'public_status' 參數是指定是否所有人可以看見這個公告的錄影檔。當 'public_status' 設定為 '1' 時，所有註冊的使用者可以看見這個公告的錄影檔。當 'public_status' 設定為 '2' 時，所有人（包括訪客）均能看見這個公告的錄影檔。'duration' 參數是指定這個公告錄影檔的持續時間（以分鐘為單位）。'title' 參數是設定錄影檔的標題。'blackbox_data' 參數是指定當使用者請求播放錄影檔時，被用來產生播放錄影檔 jnj 的可列印字串。'<user>' 項目是列出所有可以播放這個

公告錄影檔的註冊會員。`<user>` 的內容是顯示使用者的 internal_id。

28) upload file notification (MCU 3.9 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <upload_file_notification seq="0" internal_id="a" meeting_id="m" error="" utc="123456789" ip="192.168.1.1"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_response> <upload_file_notification seq="0" passed="1" errorcode="20001"/> </msgr_reponse></pre>

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

請注意，`error` 是指定上傳動作的錯誤代碼。如果上傳成功，`error` 為 `0`。`utc` 是指定由 MCU 所傳送請求的時間。

29) upload file (MCU 3.9 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <upload_file internal_id="a"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_response> <upload_file passed="1" errorcode="20001"> detailed error is written here if not passed. <meetingid>test</meetingid> </upload_file> </msgr_reponse></pre>

這個請求要求一個可被用來置放上傳錄影檔的會議編號。可以使用與建立新會議一樣的準則來產生這個會議編號。

30) file status notification (MCU 3.9 版以上)

request.xml :
<pre><?xml version="1.0" encoding="utf-8" ?> <msgr_request> <file_status_notification seq="0" internal_id="a" meeting_id="m" file_name="" action="" param="" utc="123456789"/> </msgr_request></pre>
response.xml :
<pre><?xml version="1.0" encoding="utf-8" ?></pre>


```
<msgr_response>
  <file_status_notification seq="0" passed="1" errorcode="20001"/>
</msgr_reponse>
```

請注意，`seq` 參數是非必要的，它是在批次請求時做為識別獨立請求所使用的。如果在請求中存在 `seq`，這個參數在回應中必須原封不動的重複呈現。

當 MCU 變更錄影檔狀態時，MCU 會傳送一個通知給第三方應用程式。`action` 是指定變更的動作。而 `param` 需要完全描述變更的內容。`action` 可以是 "archive"、"delete"、"mark_read" (param 其值非零表示新的狀態為已讀)、"share"、"unshare"、"jnrpassword" (param 其值非零表示檔案是受密碼保護的) 與 "jnrtitle" (`param` 顯示新的 JNR 標題)。

31) share file (MCU 3.9 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <share_file internal_id="a" meeting_id="m" file_name="f" expire=""
password="" target_internal_id=""/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <share_file passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <url> base64 encoded url </url>
    <data> printable string </data>
  </share_file>
</msgr_reponse>
```

在這個請求中，`expire` 是指定共享錄影檔網址/資料的有效天數。如果 `expire` 為 0，共享將永遠有效。如果第三方應用程式不支援有效期，它應該返回一個錯誤。`password` 是指定播放者必須進行輸入以存取共享錄影檔的一組密碼。如果第三方應用程式在共享錄影檔網址中不支援密碼，它應該返回一個錯誤。`password` 若為空字串則是表示沒有密碼限制。`target_internal_id` 是指定與目標 `internal_id` 一致的使用者才能播放共享錄影檔。這個資訊應該儲存在返回的 `data` 部份。如果第三方應用程式不支援此功能，它應該返回一個錯誤。對匿名共享來說，`target_internal_id` 必須是空字串。

在回應中的 `data` 必須是可列印字串 (如果原始資料是二進制，則使用十六進制或 base64 的編碼)。它的最大長度是 4096 (包括結尾的空值)。當使用者嘗試播放這個共享錄影檔時，這個可列印字串會提供給第三方應用程式。

更多有關共享錄影檔與取消共享錄影檔的資訊，請見本文後續的 4.9 章節。

32) view_record (MCU 3.9 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <view_record internal_id="a" meeting_id="m" file_name="f" />
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <view_record passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <url> base64 encoded url </url>
  </view_record>
</msgr_reponse>
```

這個請求要求一個列出指定檔案的共享播放記錄網址。

33) playback_file (MCU 3.9 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <playback_file internal_id="a" meeting_id="m" file_name="f" />
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <playback_file passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </playback_file>
</msgr_reponse>
```

這個請求要求一個可以播放指定錄影檔的 jnj。

34) playback_share_file (MCU 3.9 版以上)

request.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_request>
  <playback_share_file internal_id="a" blackbox_data=""/>
</msgr_request>
```

response.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<msgr_response>
  <playback_share_file passed="1" errorcode="20001">
    detailed error is written here if not passed.
    <jnj>base64 encoded jnj file content</jnj>
  </playback_share_file>
</msgr_reponse>
```

這個請求要求一個經由 blackbox data 識別可以播放共享或公告錄影檔的 jnj。

4.5 給第三方應用程式使用的線上資訊

MCU 寫入附加的使用者線上資訊至 `_status_ip_xxx.xml` 狀態檔中，更多資訊請見本文前述的 3.4 章節。第三方應用程式可以利用這個附加的使用者線上資訊來顯示使用者的線上狀態在其網頁上。

附加的使用者線上資訊使用新的標籤 `<msgr_directory>`。下列是個範例：

```
<msgr_directory>
  <user name="a" userid="3" online="true" messenger_status="1"
  office_status="0"/>
</msgr_directory>
```

每個 `'user'` 列為一位使用者列。`'name'` 與 `'userid'` 顯示該使用者的名稱與帳號。`'online'` 標籤的值是 `'true'` 或 `'false'`，用來表示使用者是否上線。詳細的資訊寫入在 `'messenger_status'` 與 `'office_status'` 參數中。`'messenger_status'` 的含意如下表所示：

1	線上
2	忙碌
3	馬上回來
4	離開
5	顯示為離線
6	線上 (行動版)
7	線上 (瀏覽器版)

下列的線上狀態計算是基於 `'messenger_status'` 與 `'office_status'` 組合的建議：

```
if(messenger_status == ONLINE_STATUS_APPEAROFF)
  "Offline";
else if(messenger_status == ONLINE_STATUS_BUSY)
  "Busy";
else if(messenger_status == ONLINE_STATUS_RIGHTBACK)
  "Be Right Back";
else if(messenger_status == ONLINE_STATUS_AWAY)
  "Away";
else if(office_status == 2)
  "In Meeting";
else if(office_status)
  "Office Open";
else if(messenger_status == 0)
  "Offline";
else if(messenger_status == ONLINE_STATUS_MOBILE)
```

```
    "Mobile";
else if(messenger_status == WEB)
    "Web"
else
    "Online";
```

4.6 Messenger 負載資訊

Messenger 負載資訊是儲存在下列狀態檔中

```
_cluster_messenger_connection_192.168.1.1_.txt
```

注意，在負載狀態檔的 IP 位址後方有個 "_" 符號。

負載狀態檔內含一行以空格分隔的三組數字，代表 "Messenger 連線數"、"訪客的 Messenger 連線數"、"Messenger 連線數的上限"。

例如，某臺伺服器在其狀態檔中可能是

```
3563 24 5000
```

這是表示這個 Messenger 伺服器目前擁有 3563 個 Messenger 連線與 24 個訪客 Messenger 連線，而其上限是 5000 個 Messenger 連線。

4.7 在 Messenger 伺服器與會議伺服器之間

要服務大規模的使用者，不同的 MCU 伺服器可以分別被設計為 Messenger 服務主機與會議服務主機。由於 Messenger 服務 MCU (即 Messenger 伺服器) 必須存取會議服務 MCU (即會議伺服器) 所產生的會議狀態檔，因此當 Messenger 服務 MCU 與會議服務 MCU 獨立運作時，它們應該在各自的 config.ini 檔中指定相同且可共同存取的錄影檔目錄。

然而，如果這兩種伺服器無法設置成相同且可共同存取的錄影檔目錄，Messenger 服務 MCU 可以採用另外一種方法來存取會議服務 MCU 所產生的會議狀態檔。在 Messenger 服務 MCU 的 config.ini 檔中，'meeting_server_ip' 設定包含所有會議伺服器的 IP 位址。Messenger 伺服器將會發起並與每一個會議伺服器保持一個 TCP 連線。會議伺服器經由這個 TCP 通道傳送會議狀態資訊給 Messenger 伺服器。為了安全起見，會議伺服器可以使用 'mmc_messenger_server_allow' 來設定 IP 位址範圍以限制可以與其連線的 Messenger 伺服器。

如果在 Messenger 伺服器與會議伺服器之間有防火牆，某些代理伺服器可能需要調整設定來

支援這個 TCP 連線。

4.8 第三方應用程式與 Messenger 伺服器之間的互動 (由第三方應用程式發起的互動)

自 MCU 3.6.0 版起，第三方應用程式可以經由 Messenger 伺服器的內部網頁服務介面來傳送某些通知給 Messenger 伺服器。

Messenger 伺服器，即 Messenger 服務 MCU，在 **http://mcuip:mcuport/** 支援一個內部網頁服務。

當第三方應用程式啟動時，它應該拜訪 **http://mcuip:mcuport/web2mcu/init** (如果可能的話)，以便觸發 Messenger 伺服器來傳送 'command list' 請求給第三方應用程式。這個請求是非必要的，因為 Messenger 伺服器每天會定期請求一次，以防萬一第三方應用程式不支援這個拜訪。

如果有任何預約活動變更，例如開始時間、持續時間或與會者變更，第三方應用程式應該拜訪 **http://mcuip:mcuport/web2mcu/event** 接著 Messenger 伺服器將被觸發來傳送一個 'event update' 給第三方應用程式。

如果有任何公佈的錄影檔變更，第三方應用程式應該拜訪 **http://mcuip:mcuport/web2mcu/publish_file** 接著 Messenger 伺服器將被觸發來傳送一個 'publish file update' 給第三方應用程式。

如果有使用者變更他的喜好設定 (例如是否所有人可以看見他)，第三方應用程式應該拜訪 **http://mcuip:mcuport/web2mcu/preference?internal_id=abc** 接著 Messenger 伺服器將被觸發來獲取使用者帳號為 'abc' 更新後的喜好設定資訊。

如果有使用者變更他的姓名，第三方應用程式應該拜訪 **http://mcuip:mcuport/web2mcu/user_update?internal_id=abc** 接著 Messenger 伺服器將被觸發來獲取使用者帳號為 'abc' 更新後的姓名資訊。

請注意，當在系統中擁有多個 Messenger 伺服器時，第三方應用程式應該拜訪相對應連結的所有 Messenger 伺服器。

4.9 共享與取消共享錄影檔

自 MCU 3.9.0 版起，MCU 支援共享與取消共享錄影檔。為了與 MCU 相容，第三方應用程

式必須滿足下列要求。否則，當使用者請求共享或取消共享錄影檔時，第三方應用程式應該返回一個錯誤訊息。

1. 當一位使用者請求播放共享錄影檔時，但相對應 xml 檔的 'shared' 旗標被重置 (請見 3.5 章節)，第三方應用程式應該返回一個錯誤訊息給使用者。
2. 當第三方應用程式共享一個檔案時，應該在相對應的 xml 檔中設定 'shared' 旗標。
3. 當第三方應用程式取消共享一個檔案時，應該在相對應的 xml 檔中重置 'shared' 旗標。

4.10 MCU 叢集

自 MCU 3.12.0 版起，MCU 支援叢集，即多個 (最高至 32) MCU 可以聚集在一起來支援大規模 (最高至 100,000) 的 Messenger 使用者。要設定 MCU 叢集，請按照下列說明。

MCU 伺服器的授權碼必須要有啟用 'clustering' 才能使用叢集功能。所有的 MCU 伺服器必須使用相同且可共同存取的 '錄影檔' 目錄，並且使用相同的金鑰對 (請見 3.8 章節)。在錄影檔目錄中建立一個列舉所有叢集 MCU 伺服器 IP 位址的純文字檔 'msl.txt'：一行一組 IP。在每個 MCU 伺服器的 configm.ini 檔中加入一個參數 'self_messenger_server_ip' 來告訴 MCU 它所屬的 IP。這個 IP 也必須是在 msl.txt 檔中的其中一行。下列是個 msl.txt 檔範例：

```
192.168.1.8
```

```
192.168.1.119
```

請注意，如果您是使用需要連結接口位址的本機 IPv6 位址，將 msl.txt 複製到一個擁有執行權限的目錄中 (與 configm.ini 同目錄)，這樣您就可以追加連結接口到 IPv6 位址。當 MCU 在擁有執行權限的目錄中找到 msl.txt 檔時，在錄影檔目錄中的 msl.txt 將被忽略。位於錄影檔目錄的 msl.txt 檔中是不允許 IPv6 位址的接口。舉例來說，如果 MCU1 使用 IP fe80::213:2ff:fe20:2454 而 MCU2 使用 IP fe80::213:2ff:fe20:2455，在 MCU1 目錄中的 msl.txt 檔內容將是

```
fe80::213:2ff:fe20:2454%2
```

```
fe80::213:2ff:fe20:2455%7
```

而在 MCU2 目錄中的 msl.txt 檔內容將是

```
fe80::213:2ff:fe20:2454%5
```

```
fe80::213:2ff:fe20:2455%9
```

基於效能考量，在叢集 MCU 伺服器之間的通訊並沒有進行加密。MCU 伺服器之間必須能夠互相直接連線且不使用任何代理伺服器設定。

第三方應用程式可以透過控制使用者端 `getmcuinfo` 請求的回應（請見 4.2 章節）來控制 Messenger 使用者端要連線的 MCU 伺服器。如果第三方應用程式決定採用負載平衡策略來分配 MCU 的工作，它可以讀取伺服器的狀態檔來確定伺服器的負載（請見 4.6 章節）。