使用 IBM Lotus Sametime Connect 開發具備視訊會談功能的外掛程式

級別：中級

IBM 技術啓用專家，Hiroaki Komine (HKOMINE@jp.ibm.com)

2008 年 10 月 28 日

IBM® Lotus® Sametime® 8.0 運用電話系統及影音功能，進一步擴大即時通訊範疇。同時，針對以 Eclipse 外掛程式架構爲基礎，本產品還提供了高度彈性平台。本文介紹 Lotus Sametime Telephony 用戶端工具集，以便在 Lotus Sametime Connect 上開發新的外掛程式。

IBM Lotus Sametime 致力發展 IBM Lotus 協同軟體產品的即時協同作業功能，已爲期長達十年。1998 年第一次發行時是可調式、安全且即時的通訊環境，也是企業協同作業的必備工具。

Lotus Sametime 7.5 則是第一個商用產品，即在 Eclipse 智慧型用戶端平台 (RCP) 上部署了即時的協同作業技術。由於 Lotus Sametime 7.5 是以 Eclipse RCP 爲基礎，您可透過 Eclipse 外掛程式架構輕鬆新增功能，也可以擴充前版 Lotus Sametime 用戶端的功能。

Lotus Sametime 7.5 及 7.5.1 有其他新的影音通訊加強功能。語音會談就是 Lotus Sametime 7.5 的新功能，使用者可以跟其他 Lotus Sametime Connect 使用者進行語音通訊。Lotus Sametime 7.5.1 的新功能則是視訊會談，可與 Lotus Sametime Connect 使用者進行視訊通訊。Lotus Sametime 使用者可在 Lotus Sametime Connect 用戶端上，使用這些標準的影音通訊功能。此外，您也可以在 Lotus Sametime Connect 用戶端及 IBM Lotus Expeditor 用戶端上執行的應用程式中使用這些功能。有了這些加強功能，Lotus Sametime 即成了 IBM Unified Communication and Collaboration (UC2) 策略的核心產品。

如今，2007 年 11 月發行的最新版 Lotus Sametime 8.0，在產品功能及延伸性方面提供更多額外的加強套件。本文透過使用 Lotus Sametime 電話系統 API 的範例程式說明 Lotus Sametime 電話系統，協助您在 Lotus Sametime Connect 用戶端上開發自己的影音應用程式。本文的範例程式爲使用 Lotus Sametime 8.0 進行開發，且預計可在 Lotus Sametime 7.5.1 上執行。

Lotus Sametime 電話系統服務

Lotus Sametime 電話系統服務包括影音會談功能及點選通話 (click-to-call) 功能。

語音會談及視訊會談是本產品的標準功能。Lotus Sametime Connect 用戶端則包括 GIPS (Global IP Solution) 授權的語音及視訊引擎，同時可建立 Lotus Sametime Connect 用戶端之間的點對點語音和視訊階段作業。該語音和視訊通訊引擎可由使用 Telephony Service Provider Interface (Telephony SPI) 的其他服務提供者取代。

點選通話功能是 Lotus Sametime 伺服器上整合企業 PBX 或電話會議的技術，可提供電話控制功能。Lotus Sametime Connect 使用者只要在 Sametime Connect 用戶端上執行操作，便可撥打電話或開會。此外，這個功能也會向 Sametime Connect 用戶端使用者發出來電通知，並以新視窗顯示來電者的資訊。

透過 Lotus Sametime 伺服器及企業 PBX 與電話會議系統之間的伺服器端整合模組，提供點選通話功能。Lotus Sametime 伺服器可發佈 PBX 供應商等服務提供者使用的 Telephony Conference Service Provider Interface (TCSPI)，以開發整合模組（服務提供者模組）。服務提供者模組可連接 Lotus Sametime 伺服器及 PBX，或電話會議系統。

除了這些影音會談及點選通話功能之外，Lotus Sametime Client Telephony 應用程式設計介面 (API) 也已加入 Lotus Sametime Connect Toolkit。您可以使用 Lotus Sametime Connect 用戶端及 Lotus Expeditor 用戶端上的 Lotus Sametime 語音與視訊功能，開發新的應用程式。

圖 1 爲視訊會談的使用者介面。

圖 1. 視訊會談使用者介面

Lotus Sametime 電話系統的架構

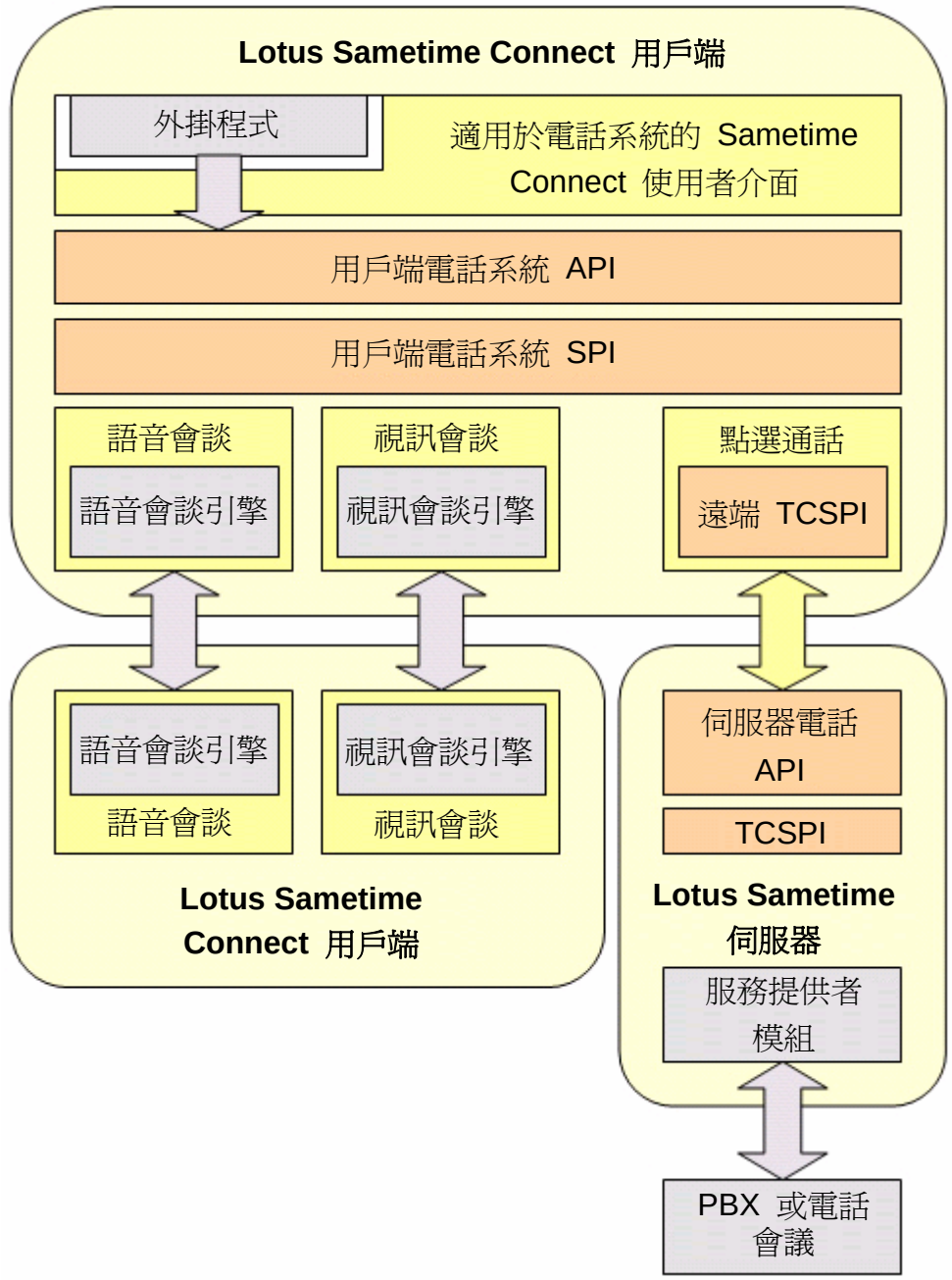
圖 2 爲 Lotus Sametime Connect 用戶端及 Lotus Sametime 伺服器上的 Lotus Sametime 電話系統服務架構。點選通話服務包括用戶端外掛程式及伺服器端服務提供者模組。用戶端遠端 TCSPI 外掛程式是 Lotus Sametime 產品隨附的元件，負責處理有關用戶端使用者介面的特定任務。在伺服器端，伺服器電話系統服務可提供 Lotus Sametime 伺服器上的點選通話服務，並且接受來自用戶端外掛程式的點選通話要求。根據來自用戶端的要求，電話系統服務可使用 TCSPI 呼叫服務提供者模組，並且控制 PBX 或電話會議系統。

另一方面，影音會談服務包括僅 Lotus Sametime Connect 用戶端使用的用戶端外掛程式。這些外掛程式就是圖 2 的語音會談 (Voice Chat) 及視訊會談

(Video Chat)。雖然 Lotus Sametime 產品有隨附影音會談實作，但您也可以開發本身的影音會談引擎實作，以取代現有的版本。如果您選擇開發本身的實作，即可使用用戶端 SPI。有了新的影音會談外掛程式實作，Lotus Sametime 隨附的現有外掛程式就會自動停用。

Lotus Sametime Connect Toolkit 會針對所有電話系統功能，包括點選通話、語音會談及視訊會談，顯示用戶端電話系統 API，以便您開發本身的使用者介面應用程式。本文根據 Lotus Sametime 產品內建的視訊會談服務，逐步說明步驟，用以開發視訊會談應用程式及新增使用者介面。

圖 2. Lotus Sametime 電話系統的架構

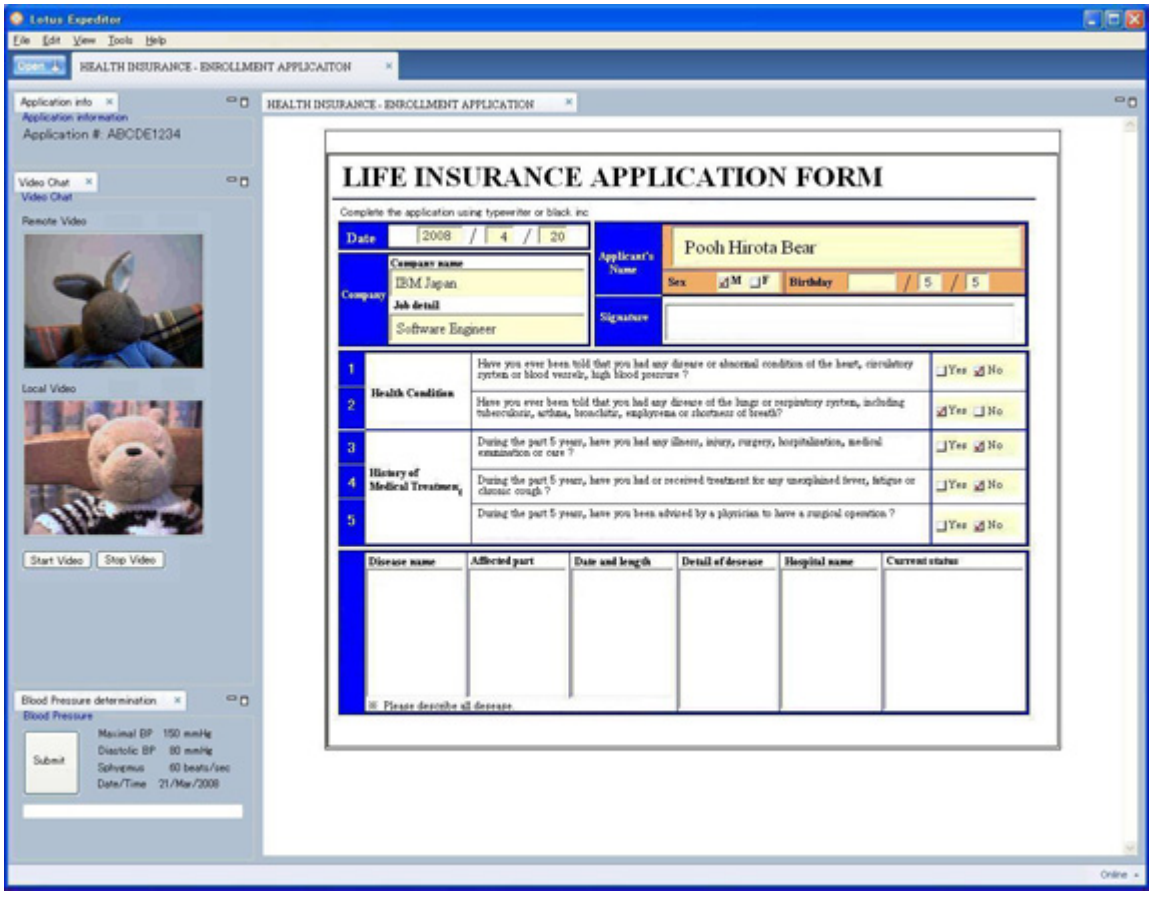


建立視訊會談應用程式

針對如何使用 Lotus Sametime 視訊會談功能，在 Lotus Sametime Connect 用戶端上開發外掛程式，本文示範了一般步驟。本文所提供的範例應用程式是執行特定互動應用程式（例如線上遊戲及遠端客戶支援）的原型，只要二人以上即可在程式中使用影音通訊分享資料及聊天。圖 3 是一個客戶支援應用程式範例，說明壽險代理員在單一 Lotus Expeditor 用戶端視窗上跟客戶會談及瀏覽 Web 應用程式頁面的情況。

這種應用程式可能需要特殊的使用者介面設計，例如啓動應用程式時可自動開啓視訊會談，並且在單一應用程式視窗內建視訊會談畫面及顯示資料。本範例應用程式會執行這些要求。

圖 3. *互動應用程式範例*



本文的範例應用程式包括 Lotus Sametime 電話系統 API 中所處理的視訊會談範例實作。您可以使用此範例，根據商業需求結合其他應用程式，開發出新的應用程式。

本文說明的範例應用程式，是以 Lotus Sametime 開發即時協同應用程式的首要步驟。

您可以下載所有範例程式碼，這是一個適用於 Eclipse 專案的保存檔；請參閱本文的下載區連結。匯入保存的專案前，請先設定開發 Lotus Sametime Connect 外掛程式。有關 Lotus Sametime Connect 外掛程式開發環境的設定步驟，請參閱 Lotus Sametime 8.0 軟體開發套件中隨附的「Lotus Sametime 8.0 整合手冊」。

若要在本身的環境中測試視訊會談應用程式，則需至少兩個 Lotus Sametime 使用者 ID。如果有自己的 Lotus Sametime 伺服器，即可隨心所欲註冊測試所需的使用者數目。否則，您也可以改用 IBM Lotus Sametime 8.0 示範網站所提供的 Lotus Sametime 示範伺服器。

範例應用程式概觀

本文介紹如何開發簡單的視訊會談應用程式，該應用程式提供下列功能：

- 應用程式：在 Lotus Sametime Connect 用戶端使用者介面中顯示一個矩形區域。
- 視訊會談呼叫者：在應用程式使用者介面上指定接受者並開始視訊會談。
- 視訊會談接受者：自動接受收到的視訊會談要求。
- 在應用程式使用者介面中，內建本端畫面及遠端畫面。

圖 4 爲應用程式。應用程式使用者介面有四個按鈕。您可以使用下列兩個按鈕與「使用者 ID」(User ID) 文字欄位指定的接受者，啓動新的視訊會談，以及停止現有的視訊會談階段作業；上列兩個按鈕則可讓您開始及停止預覽本端視訊畫面，不需要連到視訊會談階段作業。

遠端視訊及本端視訊的兩個畫面，則顯示在應用程式使用者介面的底端。與對方通訊時，您與視訊會談對象的影像都會顯示在畫面上（請參見圖 4）。預覽本端視訊時，您只可以看到自己的影像，遠端畫面會停用，如圖 5 所示。

圖 4. 範例應用程式的視訊會談

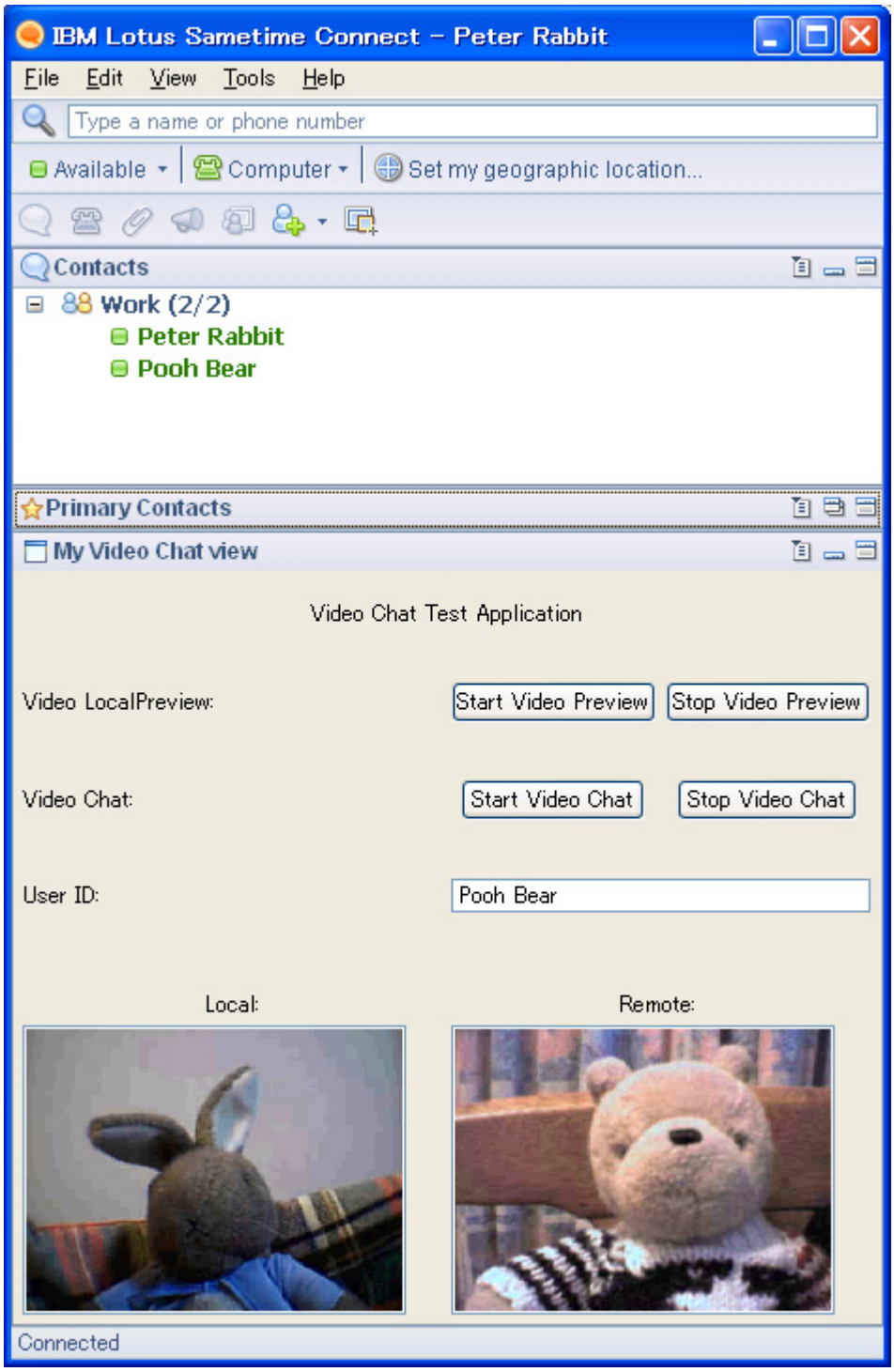
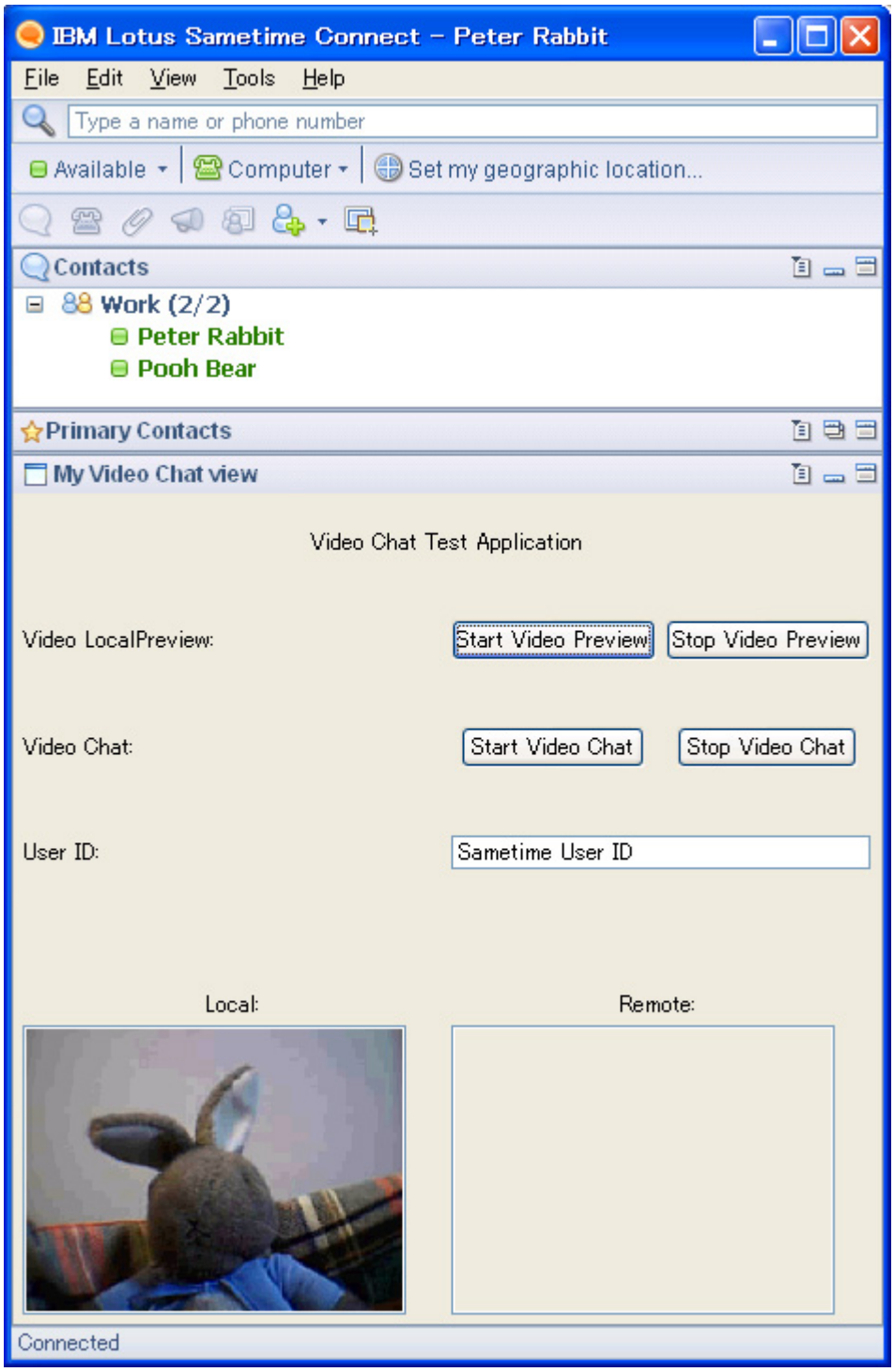


圖 5. 在範例應用程式上預覽視訊



由於本文會著重在使用電話系統 API 開發應用程式的步驟，因此建立外掛程式及新增按鈕和畫布等 SWT 元件的基本概念則不再贅述。有關擴充 Lotus Sametime Connect 用戶端的相關資訊，請參閱資源一節文章。

起始設定視訊會談

開始視訊會談的前置作業

開始視訊會談之前，必須先進行確認程序，以確定您已準備好開始視訊會談階段作業，而且對方也準備好接受視訊會談階段作業。

Lotus Sametime Connect Toolkit 隨附的 Lotus Sametime 社群會負責執行這個確認程序。請從 CommunityService 類別實例擷取 Community 類別實例，開始使用 Lotus Sametime 社群。清單 1 的範例程式碼說明如何在範例程式碼 MainView 類別的建構子中實作此實例。此程式碼遵守 Lotus Sametime Connect Toolkit 的程式設計標準。有關詳細資訊，請參閱「Lotus Sametime 8.0 整合手冊」。Lotus Sametime V8.0 Software Development Kit 會隨附此手冊。

清單 1. 執行 Lotus Sametime 服務的變數宣告

```
/*
 * variables for Sametime
 */
private CommunityService commSrv = null;

private Community community = null;

private LiveNameService lnSrv = null;

private PeopleService ppSrv = null;
```

清單 2 是用來建立社群服務物件實例的程式碼。

清單 2. 執行社群服務物件的實例化

```
/*
 * get Sametime CommunityService
 */
try {
        commSrv = (CommunityService) ServiceHub
                        .getService(CommunityService.SERVICE_TYPE);
} catch (ServiceException e) {
        e.printStackTrace();

        MessageDialog.openError(PlatformUI.getWorkbench().getDisplay()
                        .getShells()[0], "Video Start Listener Exception", e
                        .getMessage()
                        + "\nAT: " + this.getClass().toString());
```

```
            return;
}

/*
 * get Sametime Community instance
 */
community = commSrv.getDefaultCommunity();
```

使用者按一下「開始視訊會談」(Start Video Chat) 按鈕後，範例應用程式就會與「使用者 ID」(User ID) 欄位指定的接受者開始視訊會談。同時，MainView 的 Listener 內部類別 widgetSelected() 方法就會執行事件處理。

首先，必須確認您已登入 Lotus Sametime 社群。如果還沒有登入，則無法使用任何 Lotus Sametime 服務。清單 3 的程式碼會呼叫 Community 類別的 LoggedIn() 方法，並確認您已登入 Lotus Sametime 社群。如果尚未登入，就會顯示錯誤視窗。

清單 3. 執行登入狀態的確認

```
/*
 * Check if the caller has logged in.
 */
if (! community.isLoggedIn()) {
            MessageDialog.openError(comp.getShell(),
                                    "Lotus Sametime error",
                                    "Need to login Lotus Sametime");
            return;
}
```

接下來，請確認 Lotus Sametime 社群中有視訊會談呼叫接受者且作用中。否則，呼叫接受者就不會收到您的呼叫邀請。

請取得 LiveNameResolver 類別實例解析的 LiveName 類別物件，以及代表呼叫接受者的 LiveName 類別實例，以進行這些確認作業。如果 Lotus Sametime 社群有該接受者且已登入，就會針對 LiveName 類別傳回非空值。然後，從已解析的聯絡人 ID 和社群 ID 取得 Person 類別實例，並確認使用者的狀態是

STATUS_AVAILABLE。此狀態表示使用者已登入 Lotus Sametime 社群且作用中。

清單 4. 執行解析接受者及取得接受者狀態的程序。

```
/*
 * check if the participant is available on Sametime
 */
try {       /*
             * The participant must be resolved with LiveNameResolver to
             * start video chat
                      */
            lnSrv = (LiveNameService) ServiceHub
                                .getService(LiveNameService.SERVICE_TYPE);
            LiveNameResolver lnRes = lnSrv.getLiveNameResolver();
            lnRes.resolve(targetUserId);

            LiveName ln = lnSrv.getResolvedLiveName(targetUserId);
            if (null == ln) {
                      MessageDialog.openError(comp.getShell(),
                                "Sametime error",
                                "LiveName object is null for " + targetUserId);
                      return;
            }

            /*
             * get contact Id and community Id for the user
             */
            String contactId = ln.getContactId();
            String communityId = ln.getCommunityId();

            /*
             * get user online status. the participant must be available
             * to start video chat
             */
            ppSrv = (PeopleService) ServiceHub
                                .getService(PeopleService.SERVICE_TYPE);
            person = ppSrv.getPerson(contactId, communityId, false);
```

```
        if (person.getStatus() != LiveName.STATUS_AVAILABLE) {
                MessageDialog.openError(comp.getShell(),
                        "Sametime error", "Participant " + targetUserId
                                + " is not active.");
                return;
        }
} catch (Exception e) { ... }
```

開始視訊會談

現在，您已完成視訊會談的前置作業，只要在清單 5 所列的 VideoCallHandler 類別中，使用 startVideoChat() 方法，即可啓動視訊會談階段作業，並將邀請傳送給接受者。

清單 5. VideoCallHandler 類別的 startVideoChat() 方法架構

```
public static void startVideoChat(String placeId,
                                  Person targetPerson,
                                  Canvas remoteCanvas, Canvas
localCanvas)
                        throws VideoCallHandlerException {
        try {
                /*
                 * create option for video call
                 */
                 .....

                /*
                 * initiates a call using the specified options.
                 */
                 .....

                /*
                 * create address for the participant and
                 * add it to call participant
                 */
                 .....
```

```
                    /*
                     * start displaying local and
                     * remote video for this call on canvases
                     */
                     .....

        } catch (TelephonyServiceException e) {
                throw new VideoCallHandlerException(e.getMessage(), e);
        }
}
```

建立新的 Call 物件時需要某些呼叫物件，以採用該呼叫的內容值。Community ID 內容可指定 Lotus Sametime 社群，用以建立新視訊會談呼叫階段作業。在此範例程式中，此值擷取自使用者已登入的預設 Lotus Sametime 社群中。

無論 Lotus Sametime 社群中的協同作業類型爲何（包括視訊會談），系統都會爲其建立一個虛擬位置供使用者會面及分工合作。若要啓動新的視訊會談呼叫，請使用專屬位置 ID 建立一個新的虛擬位置。在本範例程式中，視訊會談的位置 ID 內容值是從 MainView 類別 generatePlaceId() 方法中的固定字串連結及亂數產生。

呼叫選項中也會設定代表視訊會談階段作業的旗標。請參閱清單 6。

清單 6. 匯入視訊會談選項實作

```
/*
 * create option for video call
 */
String communityId = CommunityUtil.getDefaultCommunity().getId();
Properties properties = new Properties();
properties.setProperty("placeId", placeId);
properties.setProperty("communityId", communityId);
CallOptions options = CallFactory.getInstance().createCallOptions();
options.setProperties(properties);
options.setVideoCall(true);
```

指定這些選項之後，程式就會從已定義的 CallManager 類別實例及事先建立的實例，以建立 Call 物件。請參閱清單 7。

清單 7. 建立呼叫管理程式物件實例實作

```
/*
 * create CallManager for Video Chat.
 */
private static CallManager manager = CallFactory.getInstance()

        .createCallManager(CallManager.SERVICE_TYPE_VIDEO_CHAT);
                .....

/*
 * initiates a call using the specified options.
 */
Call call = manager.startCall(new Vector(), options,
                                        CallListenerImpl.getInstance());
```

若要將呼叫接受者新增到視訊會談階段作業，您必須爲該接受者建立 Address 物件，然後新增到 Call 物件。本範例程式會使用 createVideoAddress() 方法，從 Person 物件建立 Address 物件。然後，呼叫 addParticipant() 方法，以便將接受者新增到呼叫階段作業。請參閱清單 8。

清單 8. 建立參與者位址物件實例實作

```
/*
 * create address for the participant and
 * add it to call participant
 */
Address address = createVideoAddress(targetPerson);
call.addParticipant(address);
```

createVideoAddress() 方法可建立新的 Address 物件，然後設定該物件的必要值。setAddressUri() 方法設定的接受者 URL 字首必須是 video:，以符合 Lotus Sametime 用戶端電話系統 API 規格。請參閱清單 9。

清單 9. 建立接受者位址物件實例的協助方法

```
private static Address createVideoAddress(Person person) {

        Address address = CallFactory.getInstance().createAddress();
        address.setAddressUri("video:" + person.getContactId());
```

```
        address.setParticipantId(person.getContactId());
        address.setParticipantName(person.getName());
        address.setCommunityId(person.getCommunityId());

        return address;
}
```

隨後，範例程式會呼叫 startVideo() 方法，以便在指定的 SWT 畫布上顯示此呼叫的本端及遠端視訊畫面。請參閱清單 10。

清單 10. 起始設定視訊會談階段實作

```
/*
 * Start displaying local and
 * remote video for this call on canvases
 */
call.startVideo(remoteCanvas, localCanvas);
```

針對起始設定視訊會談呼叫，您已完成所有步驟。現在，呼叫參與者將會收到來自您的程式，所發出的視訊呼叫邀請。

出席收到的視訊會談

準備接受邀請

若要接受收到的視訊會談邀請，必須具備視訊會談應用程式。本文的範例應用程式介紹以最簡單的方式處理收到的呼叫，即接受所有收到的視訊會談邀請。

收到的視訊會談邀請會以 Lotus Sametime 呼叫事件的方式發出，所以應用程式必須接聽這類呼叫事件。因此，CallListenerImpl 類別會定義成 CallListener 介面的實作，並設爲 CallManager 類別，作爲接聽器。然後，在本範例程式 VideoCallHandler 類別的 startListening() 方法中執行。請參閱清單 11。

清單 11. 將接聽器類別登錄到呼叫管理程式實作

```
public static void startListening() throws VideoCallHandlerException
{
        if(!isListening)
                try {
```

```
            manager.addListener(CallListenerImpl.getInstance());
                        isListening = true;
                    } catch(TelephonyServiceException e) {
                        throw new
VideoCallHandlerException(e.getMessage(), e);
                    }
}
```

若已登入 Lotus Sametime 社群，範例程式會呼叫 MainView 類別建構子中的 VideoCallHandler 類別方法 startListening()。請參閱清單 12。

清單 12. 在 MainView 類別建構子中，起始設定 VideoCallHandler 類別實作

```
/*
 * test login status and set CallListener to detect IncomingCallEvent
 */
if (community.isLoggedIn()) {
            try {
                        VideoCallHandler.startListening();
            } catch (VideoCallHandlerException e) {
                        e.printStackTrace();


            MessageDialog.openError(PlatformUI.getWorkbench().getDisplay()
                                    .getShells()[0], "Video Start
Listener Exception", e
                                    .getMessage()
                                    + "\nAT: " + this.getClass().toString());
            }
}

/*
 * in any case, set community listener to detect logged-in event
 */
commSrv.addCommunityListener(new CommunityListenerImpl());
```

由於內部會進行作業計時，所以在您登入 Lotus Sametime 社群之前，可呼叫檢視類別建構子。有鑑於此，範例程式會宣告社群接聽器類別，以接受登入的事件並呼叫 startListening() 方法。

Lotus Sametime Connect 用戶端登入後，即呼叫 handleCommunityLoginEvent() 方法，最後呼叫 startListening() 方法來新增呼叫接聽器。

清單 13. 登入 Lotus Sametime 社群後，起始設定 VideoCallHandler 類別實作

```
public void handleCommunityLoginEvent(CommunityLoginEvent ev) {

        int type = ev.getType();

        if (CommunityLoginEvent.TYPE_LOGGED_IN == type) {

                /*
                 * set call listener
                 */
                try {
                        VideoCallHandler.startListening();
                } catch (VideoCallHandlerException e) {
                        e.printStackTrace();


        MessageDialog.openError(MainView.getComposite().getShell(),
                                "VideoC Start Listening Exception",
e.getMessage()
                                                + "\nAT: " +
this.getClass().toString());
                }

        }

                .....
}
```

出席視訊會談

發出新的呼叫事件後，就會使用 CallListener 介面的 processCallEvent() 方法通知。若要接受收到的視訊會談呼叫，必須在該方法中執行適當程序。該方法必須確認此呼叫事件是收到的視訊會談呼叫事件。就此範例應用程式而言，會確認此 CallEvent object 是否爲 IncomingCallEvent 的實例，以及其 isVideoCall() 旗標是否爲真。請參閱清單 14。

清單 14. VideoCallHandler 類別的 processCalEvent() 方法架構

```
public void processCallEvent(CallEvent ev) throws
TelephonyServiceException {

        if (ev instanceof IncomingCallEvent) {

                /*
                 * get IncomingCallEvent object
                 */
                final IncomingCallEvent icEv = (IncomingCallEvent) ev;
                if (! icEv.isVideoCall()) {
                        System.out.println("As this is not video call, do
nothing");
                        return;
                }

                /*
                 * get call ID, community Id, and place ID for the incoming
call
                 */
                .....

                /*
                 * get composite object of main view to show video images
on it
                 */
                .....

                /*
                 * get local/remote canvases on which video images will be
shown
                 */
```

```
                .....

                /*
                 * call attendVideoChat() to accept the incoming call
invitation
                 */
                .....

        } else if (CallEvent.EVENT_CALL_TERMINATED ==
ev.getEventType()) {
                .....
        }

}
```

如果呼叫事件是視訊會談邀請，此範例就會從 IncomingCallEvent 物件（如社群 ID、位置 ID 及呼叫 ID）取得資訊。請參閱清單 15。

清單 15. 擷取收到的視訊會談資訊實作

```
/*
 * get call ID, community Id, and place ID for the incoming call
 */
final String communityId = icEv.getCommunityId();
final String placeId = icEv.getPlaceId();
final String callId = ev.getCallId();
```

此外這個範例也會示範，從顯示視訊畫面的 MainView 取得 SWT 使用者介面元件。請參閱清單 16。

清單 16. 擷取視訊會談的 SWT 元件實作

```
/*
 * get composite object of main view to show video images on it
 */
final Composite comp = MainView.getComposite();

/*
 * get local/remote canvases on which video images will be shown
```

```
 */
final Canvas localCanvas = MainView.getDefault().getLocalCanvas();
final Canvas remoteCanvas = MainView.getDefault().getRemoteCanvas();
```

最後，processCallEvent() 方法會呼叫 VideoCallHandler 類別的 attendVideoChat() 方法，以接受視訊會談邀請。此作業需要使用者介面元件的控制項，這些元件只能由應用程式的主要執行緒來執行。爲解決此問題，本範例程式從主視圖 Composite 物件取得 Display 物件，然後使用 syncExec() 方法呼叫 attendVideoChat() 方法，以在畫布上顯示視訊畫面。請參閱清單 17。

清單 17. 實作接受視訊會談邀請的執行緒

```
/*
 * call attendVideoChat() to accept the incoming call invitation
 */
/*
 * Although the attendVideoChat will call startVideo with Canvas
 * class, it is not allowed for another thread to control a UI
 * component. Therefore, we create another thread to request UI
 * handling in the Composite. Otherwise we will get "ViewPart
 * org.eclipse.swt.SWTException: Invalid thread access"
 */

if ((comp != null) && !(comp.isDisposed())) {
        comp.getDisplay().syncExec(new Runnable() {

                public void run() {
                        try {
                                /*
                                 * accept the incoming video chat invitation
                                 */
                                VideoCallHandler.attendVideoChat(callId,
                                                communityId, placeId,
localCanvas,
                                                remoteCanvas);

                                /*
                                 * set place ID for the incoming video chat
session
```

```
                                        */
                                       MainView.getView().setPlaceId(placeId);

                                } catch (VideoCallHandlerException e) {
                                       e.printStackTrace();

                                       MessageDialog.openError(comp.getShell(),
                                              "Video Attending Exception",
e.getMessage()
                                                                     + "\nAT: "
                                                                     +
this.getClass().toString());
                                }
                        }
                });

}
```

範例程式碼中的 VideoCallHandler 類別 attendVideoChat() 方法與 startVideoChat() 方法類似，差別在於其接受者不需要產生新的呼叫 ID 或新增其他參與者。請參閱清單 18。

清單 18. 執行 attendVideoChat() 方法，以接受視訊會談要求

```
public static void attendVideoChat(String callId,
                                              String communityId,
                                              String placeId,
                        Canvas remoteCanvas, Canvas localCanvas)
                        throws VideoCallHandlerException {
                try {

                        /*
                         * create option for video call
                         */
                         Properties properties = new Properties();
                         properties.setProperty("placeId", placeId);
                         properties.setProperty("communityId", communityId);
                         CallOptions options = CallFactory.getInstance().
                                                       createCallOptions();
```

```
                options.setProperties(properties);
                options.setVideoCall(true);

               /*
                * initiates a call using the specified options.
                */
               Call call = manager.connectIncomingCall(callId, options,
                        CallListenerImpl.getInstance());
               calls.put(placeId, call);

               /*
                * Start displaying both local and
                * remote video for this call on canvases
                */
               call.startVideo(localCanvas, remoteCanvas);

        } catch (TelephonyServiceException e) {
                        throw new
VideoCallHandlerException(e.getMessage(), e);
        }
}
```

現在，本範例應用程式即可接受收到的視訊會談（從呼叫者的範例應用程式起始設定）。

總結

Lotus Sametime 用戶端電話系統 API 的程式設計功能，可讓您使用 Eclipse 外掛程式技術爲基礎，打造即時的協同應用程式。

由於 Lotus Sametime 產品可提供高度可調整性及安全性，符合企業客戶的需求，因此 Lotus Sametime Connect 用戶端可作爲統一通訊及協同作業的企業應用程式平台。

IBM Lotus Expeditor 6.1.1 及 IBM Lotus Notes® 8.0 上內建的 Lotus Sametime 經驗，也有提供本文的相同用戶端電話系統功能，您可以使用 Lotus Sametime 的統一通訊及協同作業技術，在這些平台上開發新的應用程式。

下載

名稱	檔案大小	下載方法
y-lot-videchat-sample.zip	11.6KB	**HTTP**

下載方法的相關資訊

資源

學習

- 閱讀 developerWorks 文章：" Extending IBM Lotus Sametime Connect V7.5."
- 閱讀 developerWorks 文章：" Developing a location awareness plug-in for IBM Lotus Sametime Connect V7.5."

取得產品與技術

- 下載 Lotus Sametime Toolkit，包括程式庫、說明文件及範例程式碼。
- 下載 Lotus Sametime V8.0 Software Development Kit (SDK)。

討論

- 加入討論區。

關於作者

Hiraki Komine 是任職於東京的 IBM 技術啓用專家，專精於即時及團隊協同作業軟體。聯絡方式爲 HKOMINE@jp.ibm.com。