TurboMeeting Integration Specification for Partners

Version 3.4

This manual contains confidential information regarding RHUB web conferencing system and services. Distribution of this manual must have the written permission from RHUB.

RHUB Communications, Inc.
2953 Bunker Hill Lane
Suite 400
Santa Clara, CA 95054

May, 2007

Index

TurboMeeting Integration Specification

Version	Date	Changes
3.4	6/22/07	• Add <__JoinMeetingURL__> to Query Scheduled Meetings
3.3	6//17/07	• Add <__firstName__>, <__LastName__>, <__QueryScheduledMeetingURL__> to User Authentication • Drop URL parameters in Query Scheduled Meetings • Drop the schedule meeting button
3.2	6/13/07	• Add user_id to Schedule Meeting • Add service_company_uid to Query Scheduled Meeting and User Authentication • Drop links: “Profile”, “Edit”, “New” scheduled meeting
3.1	5/15/07	• Add “Query Customer Account Status” API • Add <__ScheduleMeetingURL__> in User Authentication

1. Overview

This document describes the interfaces to integrate with RHUB TurboMeeting Web Conferencing system. The following figure shows the internet based integration architect. The integration interfaces between TurboMeeting (TM) Server and Partner Server are based on URL calls.

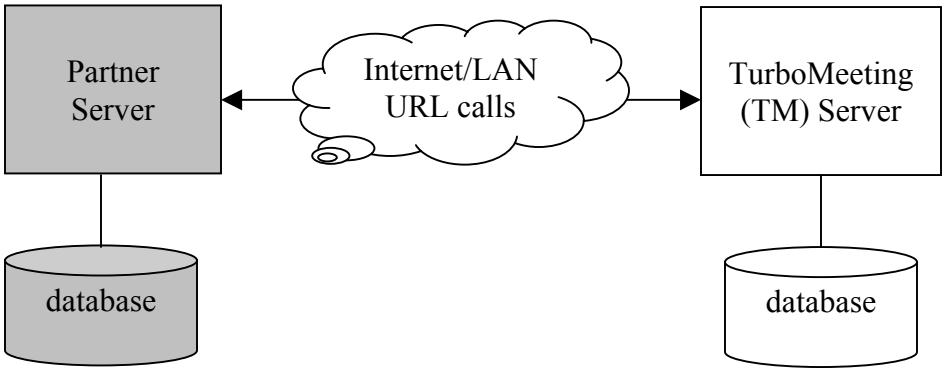


Figure 1. Integration Architect

This integration manual is for the partner that needs:

- Its own identity — private URL’s for a user to start and join meetings
- Control of user authentication
- Control of meeting scheduling activities

The following figure shows the deployment architect for this type of integration.

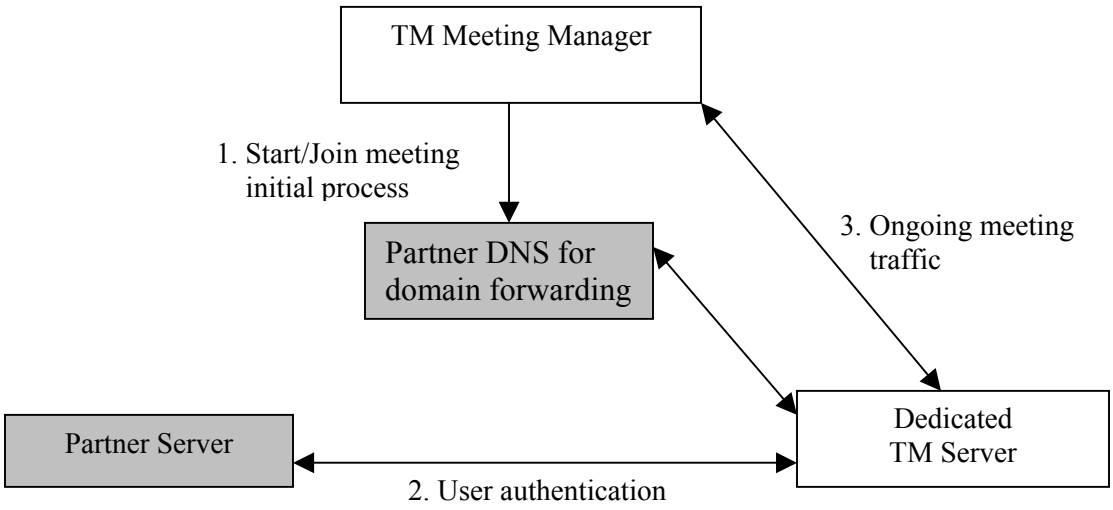


Figure 2. Deployment Architect

The table below shows the services provided by each party.

Services	**RHUB**	**Partner**
Dedicated TM Server to conduct meetings	√	
User authentication service		√
DNS service		√
Meeting scheduling services		√

The following table details the URL's and parameters that are part of the services provided by each party.

Parameters	**Reference**	**RHUB**	**Partner**
TM Server IP address		√	
Private key and algorithm to encrypt password		√	
Partner UID for TM Server to ensure service integrity		√	
URL to generate a scheduled meeting	Section 7	√	
URL's to generate and update user accounts	Section 2-4	√	
URL's to save and update users licensees	Section 2-3	√	
URL to start an unscheduled meeting	Section 8	√	
URL to join a meeting	Section 10	√	
Partner server IP address for TM Server to ensure service integrity			√
User authentication URL	Section 6		√
__QueryScheduledMeetingURL__, which must carry user session ID per login event to avoid relogin during the login session	Section 6		√

__JoinMeetingURL__, emebdded in each meeting XML return	Section 11		√
Password Assistance URL	One time setup		√
Home URL to host and join meetings via browser	One time setup		√

User Password and Meeting Password security:

To ensure the maximum security on user password and meeting passwords, here are the prinples that this integration follows:

1. Only password hashes (SHA or MD5) are transmitted over the Internet
2. Only meeting password hashes (SHA or MD5) are stored in TM server.

2. Create Customer Account

- **URL call from Partner Server to TM Server:**
  http://TMServerIP/as/wapi/service/ create_account

- **Parameters in the URL**:

Variable Name	Comment
service_company_uid	Partner ID
company_name	Customer organization name
email	Email of the main contact
number_of_meeting_room	The number (integer) of meeting rooms
number_of_participant	The number (integer) of participants per meeting room

- **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
      <__CompanyUid__>
            up to 36-character ID for the account created
      </__CompanyUid__>
</__Return__>
```

where “Status” values include:

- SUCCEED

- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution and the “company_uid” value is empty.

3. Update Customer Account

- **URL call from Partner Server to TM Server**: http://TMServerIP/as/wapi/service/update_account

- **Parameters in the URL**:

Variable Name	Comment
service_company_uid	Partner ID
company_name	Customer organization name
email	Email of the main contact
number_of_meeting_room	The number (integer) of meeting rooms
number_of_participant	The number (integer) of participants per meeting room
company_uid	The user account ID returned from “Create Customer Account”

- **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
</__Return__>
```

where “Status” values include:

- SUCCEED
- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution.

4. Delete Customer Account

- **URL call from Partner Server to TM Server:** http://TMServerIP/as/wapi/service/delete_account

- **Parameters in the URL**:

Variable Name	Comment
service_company_uid	Partner ID
company_uid	Customer ID

- **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
</__Return__>
```

where “Status” values include:

- SUCCEED
- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution.

5. Query Customer Account Status

- **URL call from Partner Server to TM Server:**
  http://TMServerIP/as/wapi/service/query_account

  - **Parameters in the URL**:

Variable Name	Comment
service_company_uid	Partner ID
company_uid	Customer ID

  - **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
      <__CompanyName__>
            Company name
      </__CompanyName__>
      <__Email__>
            Email
      </__Email__>
```

```
      <__NumberOfMeetingRoom__>
            Number of meeting rooms
      </__NumberOfMeetingRoom__>
      <__NumberOfParticipant__>
            Number of participants
      </__NumberOfParticipant__>
</__Return__>
```

where “Status” values include:

- SUCCEED
- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution. If the user account has been deleted, FAILED will be returned.

6. User Authentication

- **URL call from TM Server to Partner Server:**
  http://partner-defined-url

- **Parameters in the URL**:

Variable Name	Mandatory	Comment
email	Yes	User email
password	Yes	This is encrypted by a given private key from RHUB and the encryption algorithm
service_company_uid	Yes	Partner ID

- **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
      <__CompanyUid__>
            This must be a valid user account ID generated by TM
            Server (see Section 2)
      </__CompanyUid__>
      <__FirstName__>
            First name of the user
      </__FirstName__>
      <__LastName__>
            Last name of the user
      </__LastName__>
      <__UserID__>
            The user ID in partner server
```

```
      </__UserID__>
      <__QueryScheduledMeetingURL__>
            The URL to query the list of scheduled meetings. The
            URL should be invalid when this login user-session is
            expired. Keep the session alive for at least 20
            minutes. If this field is empty, TM client & server
            will manage the scheduled meetings.
      </__QueryScheduledMeetingURL__>
</__Return__>
```

where “Status” values include:

- SUCCEED
- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution.

7. Schedule a Meeting

- **URL call from Partner Server to TM Server:**
  http://TMServerIP/as/wapi/service/remote_schedule

- **Parameters in the URL**:

Variable Name	Mandatory	Comment
service_company_uid	Yes	Partner ID
company_uid	Yes	Customer ID
user_id	Yes	User ID

- **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
      <__MeetingID__>
            The meeting ID for the scheduled meeting
      </__MeetingID__>
</__Return__>
```

where “Status” values include:

- SUCCEED
- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution.

8. Start an Unscheduled Meeting

For the users who have downloaded TM Meeting Manager, they would most likely start a meeting by clicking the TM Meeting Manager icon on their desktop.

For the users have not download the meeting manager, Partner publishes the URL below for a user to start a unscheduled meeting

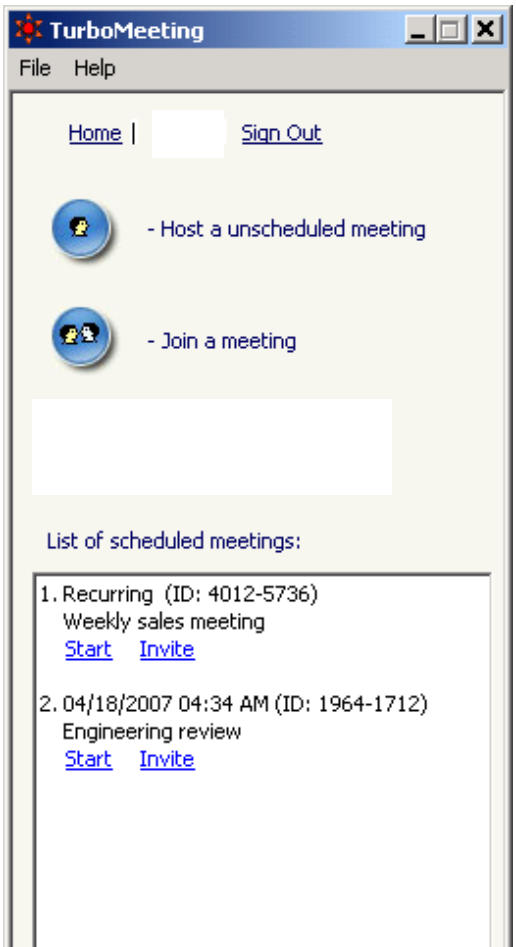
- **URL**: http://TMServerIP/as/wapi/goto_downloader?role=host

In order for a user to start a meeting, TM Server will call Partner Server to authenticate the user (see Section 5).

9. Start a Scheduled Meeting

A user starts a scheduled meeting always by using the TM Meeting Manager, which will list all current active scheduled meetings. Clicking one of the meetings, a user will start the meeting. See Figure 3.

Partner Server does not need to anything except responding the call for scheduled meeting list described in Section 10 when the user signs in the TM Meeting Manager.



10. Join A Meeting

For the users who have downloaded TM Meeting Manager, they would most likely join a meeting by clicking the TM Meeting Manager icon on their desktop.

For the users have not download the meeting manager, Partner publishes the URL below for a user to join a meeting

- **URL**: http://TMServerIP/as/wapi/goto_downloader?role=attendee

Partner Server does not need to do anything for joining meetings. However, partner does need to provider its customized join-meeting URL to RHUB in order to configure the dedicated TM Server so that the meeting host meeting panel can display the customized join-meeting URL.

11. Query Scheduled Meetings

- **URL call from TM Server to Partner Server:**
  http://partner-defined-url

  This URL is user-specific, which is the value of `<__QueryScheduledMeetingURL__>` returned from User Authentication.

- **Return**

```
<__Return__>
      <__Status__>
            The status of execution
      </__Status__>
      <__Reason__>
            The reasons to explain failed execution
      </__Reason__>
      <__MeetingList__>
         <__Meeting__>
```

```
            <__MeetingId__>
               The scheduled meeting ID, which is acquired in
               Section 6.
            </__MeetingId__>
            <__ScheduledStartTime__>
               The scheduled start time (YYYY/MM/DD HH:MM)
            </__ScheduledStartTime__>
            <__MeetingTopic__>
               The meeting subject
            </__MeetingTopic__>
            <__JoinMeetingURL__>
               The URL for attendees to join the meeting
            </__JoinMeetingURL__>
         </__Meeting__>
      </__MeetingList__>
</__Return__>
```

where “Status” values include:

- SUCCEED
- FAILED

If the Status is FAILED, “Reason” value will show the reasons of the failed execution.