

Polycom RMX System External Database API Programmer's Guide

Version 2.0

POLYCOM

Copyright © 2007 Polycom, Inc.
All Rights Reserved

Catalog No. eDOC2277B
Version 2.0

Proprietary and Confidential
The information contained herein is the sole intellectual property of Polycom, Inc. No distribution, reproduction or unauthorized use of these materials is permitted without the expressed written consent of Polycom, Inc. Information contained herein is subject to change without notice and does not represent commitment of any type on the part of Polycom, Inc. Polycom and Accord are registered trademarks of Polycom, Inc.

Notice
While reasonable effort was made to ensure that the information in this document was complete and accurate at the time of printing, Polycom, Inc., cannot assume responsibility for any errors. Changes and/or corrections to the information contained in this document may be incorporated into future issues.
Portions, aspects and/or features of this product are protected under United States Patent Law in accordance with the claims of United States Patent No: US 6,300,973; US 6,496,216; US 6,757,005; US 6,760,750; and US7,054,820. PATENT PENDING

Regulatory Notices

United States Federal Communication Commission (FCC)

Part 15: Class A Statement. This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. Test limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manuals, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his or her own expense.

United States Safety Construction Details:

- Unit is intended for RESTRICTED ACCESS LOCATION.
- Unit is to be installed in accordance with the National Electrical Code.
- The branch circuit overcurrent protection shall be rated 20 A for the AC system.
- This equipment has a maximum operating ambient of 40°C, the ambient temperature in the rack shall not exceed this temperature.

To eliminate the risk of battery explosion, the battery should not be replaced by an incorrect type. Dispose of used batteries according to their instructions.

CE Mark R&TTE Directive

Polycom Inc., declares that the Polycom RMX 2000 is in conformity with the following relevant harmonized standards:

EN 60950-1:2001

EN 55022: 1998+A1:2000+A2:2003 class A

EN 300 386 V1.3.3: 2005

Following the provisions of the Council Directive 1999/CE on radio and telecommunication terminal equipment and the recognition of its conformity.

Canadian Department of Communications (EC)

This Class [A] digital apparatus complies with Canadian ICES-003.

Notice: The Industry Canada label identifies certified equipment. This certification means that the equipment meets telecommunication network protective, operational and safety requirements as prescribed in the appropriate Terminal Equipment Technical Requirements document(s). The Department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations. Repairs to certified equipment malfunctions, may give the telecommunications company causes to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Caution: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

Regulatory Notices

Chinese Communication Certificate

声　明

此为 A 级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，可能需要用户对其干扰采取切实可行的措施。

Table of Contents

Introduction 1-1
The RMX Authorization Mechanism 1-3
User Authorization Verification 1-3
Participant Authorization Verification 1-3
Using an External Database Application to Verify Participant Rights to Create an Ad Hoc Conference 1-3
Using an External Database Application to Verify Participant Rights to Connect to the Conference 1-4
Defining Participant Rights in an External Database 1-4
Using an External Database Application to Provide Conference and/or Participant Information 1-5
Conference Information 1-5
Participant Information 1-5
External Database Application Implementation Scenarios 1-6
Data Flow between the Entry Queue, the Conference IVR and the External Database Application 1-6
Scenario 1: Verifying the Participant Rights to Create a New Conference 1-7
Scenario 2: Verifying the Participant Rights to Create a New Conference and Access the Conference 1-8
Scenario 3: Verifying Participant Rights to Create a New Conference and to Join the Conference as Chairperson 1-9
RMX Configuration for External Database Application Authorization 2-1
Configuring the MCU to use an External Database Application 2-2
Enabling External Database Validation for Starting New Ongoing Conferences 2-4
Using the RMX Web Client to Configure Entry Queue IVR Services 2-4
Using the RMX API to Configure Entry Queue IVR Services . 2-5
Enabling External Database Validation for Conferences Access 2-6
Using the RMX Web Client to Configure Conference IVR Services 2-7

Using the RMX API to Configure Conference IVR Services .. 2-10
Procedure 1 - Configuring the Conference IVR Service to access an external database application to verify conference entry or chairperson passwords 2-10
Procedure 2 - Configuring the Conference IVR Service to prompt the participant to enter the conference entry password 2-11
Procedure 3 - Configuring the Conference IVR Service to prompt the participant for the chairperson identifier key and chairperson password 2-12
Working with the External Database API 3-1
About the XML Requests and Responses 3-1
The Syntax of the XML Requests and Responses 3-3
The Create Conference XML 3-3
The Create Conference Request 3-3
The Create Conference Response 3-5
The Add Participant XML 3-10
The Add Participant Request 3-10
The Add Participant Response 3-13
The Login XML 3-16
The Login Request 3-16
The Login Response 3-18
The Keep Alive XML 3-20
The Keep Alive Request 3-20
The Keep Alive Response 3-21
Using the Sample Scripts 4-1
About the Sample Scripts 4-1
System Requirements 4-1
Installation and Configuration Procedures 4-2
Creating a Virtual Directory in IIS 4-3
Defining Security Properties for the Directory Configuration Files 4-8
Configuring the Directory Configuration File 4-16
Testing the Directory Configuration File using the External Database Simulator 4-22

1

Introduction

The Polycom RMX system Ad Hoc conferencing feature enables participants to start ongoing conferences on-the-fly, without prior definition. All participants can be allowed to start Ad Hoc conferences, or alternatively, the MCU can use an external database application to verify participant rights to initiate a new conference. In addition, the MCU can use an external database application to verify participant rights to join a specific ongoing conference. The MCU can also use an external database application to verify that an RMX user is authorized to log into the MCU.

The external database application can be the MGC WebCommander Web Server, or any other database server you choose, for example, a database server that interfaces with an existing company database. In addition to verifying participant rights to start a conference, the external database application can provide the MCU with other information from the database, such as the conference password or the billing code. Similarly, when verifying participant rights to join a conference, the external database application can provide the MCU with participant details. When verifying that a user is authorized to log in to the MCU, the external database application can provide the MCU with the user's authorization level.

All communication between the MCU and the external database application is performed using the Polycom RMX system External Database API. The MCU sends a Create Conference XML request, an Add Participant XML request or a Login XML request, as applicable, to the external database application, and the application must return an XML response verifying whether the participant can create or join the conference, or whether the user can log in to the MCU.



The Polycom ReadiRecorder can also work with an external database application. Sites can choose to work with two databases, for example, to use an external database for users, and the internal database for languages.

This guide only describes the External Database API requests and responses that are applicable to standard MCUs. The Recording request and response used by ReadiRecorder systems are not documented in this guide.

To simplify implementation, sample scripts are provided which can be used instead of the raw API. The scripts can be used to communicate with any directory type database. The user only has to provide the mapping between the directory attributes and the XML elements, using the supplied scripts.

This guide describes the format of the XML requests sent by the MCU to the external database application, and the format of the responses that the external database application must send back to the MCU. The guide also provides instructions for working with the supplied scripts.



The MCU can only work with one external database application.

The RMX Authorization Mechanism

User Authorization Verification

Users are defined in the MCU, and are assigned a user name, password and authorization level. An external database application can be used to verify that a specific user is authorized to log in to the MCU with a specific password. In the external database you must define all users who are authorized to log in to the MCU. For each user defined in the database, you must enter the user name, password, and optionally, the authorization level. The user authorization is checked in the MCU first, and then in the external database.

Participant Authorization Verification

The RMX Ad Hoc conferencing feature enables participants to start ongoing conferences on-the-fly, without prior definition when dialing an Ad Hoc-enabled Entry Queue. The created conference parameters are taken from the Profile assigned to the Ad Hoc-enabled Entry Queue. Profiles can only be accessed by users with Operator permission.

Using an External Database Application to Verify Participant Rights to Create an Ad Hoc Conference

Ad Hoc conferencing is available in two modes:

- Ad Hoc Conferencing without Authentication

  Any participant can dial into an Entry Queue and initiate a new conference if the conference does not exist. This mode is usually used for the organization's internal Ad Hoc conferencing.
- Ad Hoc conferencing with external database authentication

  In this mode, the participant's right to start a new conference is validated against a database.

This guide contains only the information about Ad Hoc conferences that is relevant to users of the RMX External Database API. For complete information about Ad Hoc conferences, refer to the *RMX Administrator's Guide*.

Using an External Database Application to Verify Participant Rights to Connect to the Conference

The external database application can also be used to validate the participant's right to join an ongoing conference. Conference access authentication can be:

- Part of the Ad Hoc conferencing flow where the participants must be authorized before they can enter the conference created in the Ad Hoc flow. It is recommended that the conference entry and/or chairperson passwords are returned by the external database application, together with the authorization to start the new conference.
- Independent of Ad Hoc conferencing where conference access is validated for conferences running on the MCU regardless of the method in which the conference was started.

The RMX prompts the user to enter the conference entry or chairperson password as relevant, and will then verify with the external database application whether the participant has the right to join the conference, and whether he/she is the conference chairperson.

Defining Participant Rights in an External Database

The external database application can be the MGC WebCommander Web Server, or any other database server you choose, for example, a database server that interfaces with an existing company database. In the external database you must define all participants with rights to start a new conference using Ad Hoc conferencing. For each participant defined in the database, you enter the conference ID, and any of the following information as required: the conference entry password, the conference chairperson password, the billing code, the participant's PIN code and his/her VIP status, and conference general information (corresponding to the contents of the Conference Info fields).

The external database application should implement company validation requirements. For example, when participants enter their PIN code when prompted for the conference or chairperson password, the application will verify the PIN code in the external database.

Using an External Database Application to Provide Conference and/or Participant Information

Conference Information

For Ad Hoc conferences, when an external database application is used to verify participant rights to create a conference, the external database application can return the following conference information:

- The conference name.
- The conference entry and chairperson passwords.
- The conference billing code.
- The conference owner.
- The maximum number of participants allowed for the conference.
- The minimum number of participants for whom resources should be reserved. With the RMX, default value is 0.
- The contents of the User Defined fields for the conference, that is, conference information to be written to the CDR file, such as the company name or telephone number.



The maximum and minimum number of participants are not returned when the WebCommander Web Server is used as the external database application.

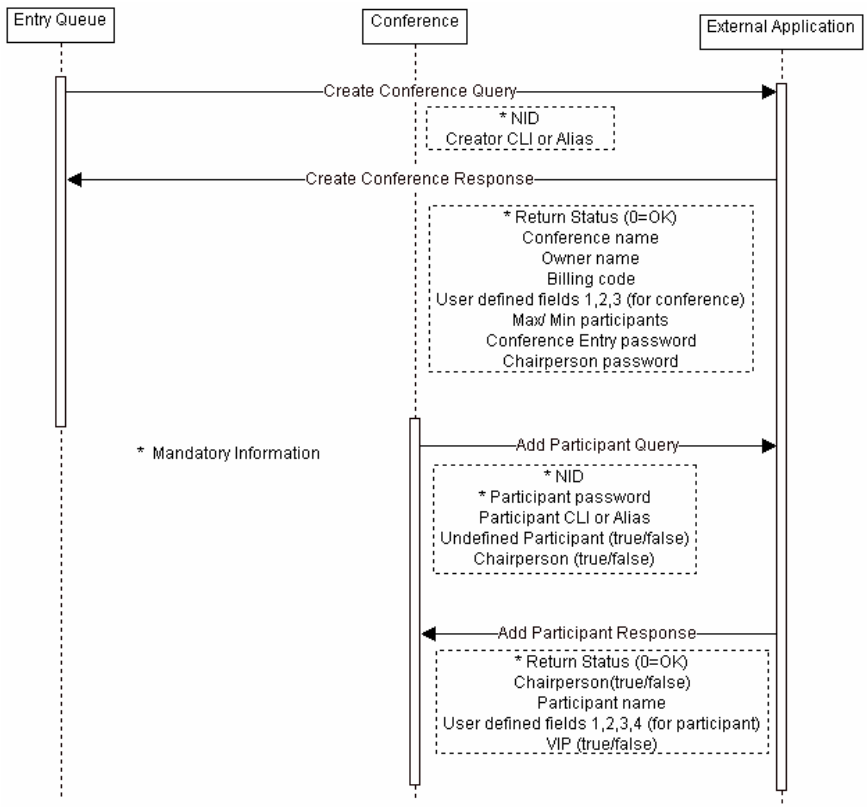
Participant Information

When an external database application is used to verify conference entry or chairperson passwords before allowing a participant to join a conference, the external database application can return the following participant information:

- The participant's display name.
- Whether or not the participant is a VIP. *This value is not supported in the RMX.*
- The contents of the Participant Info fields, that is participant information to be written to the CDR file, such as the participant's e-mail address.

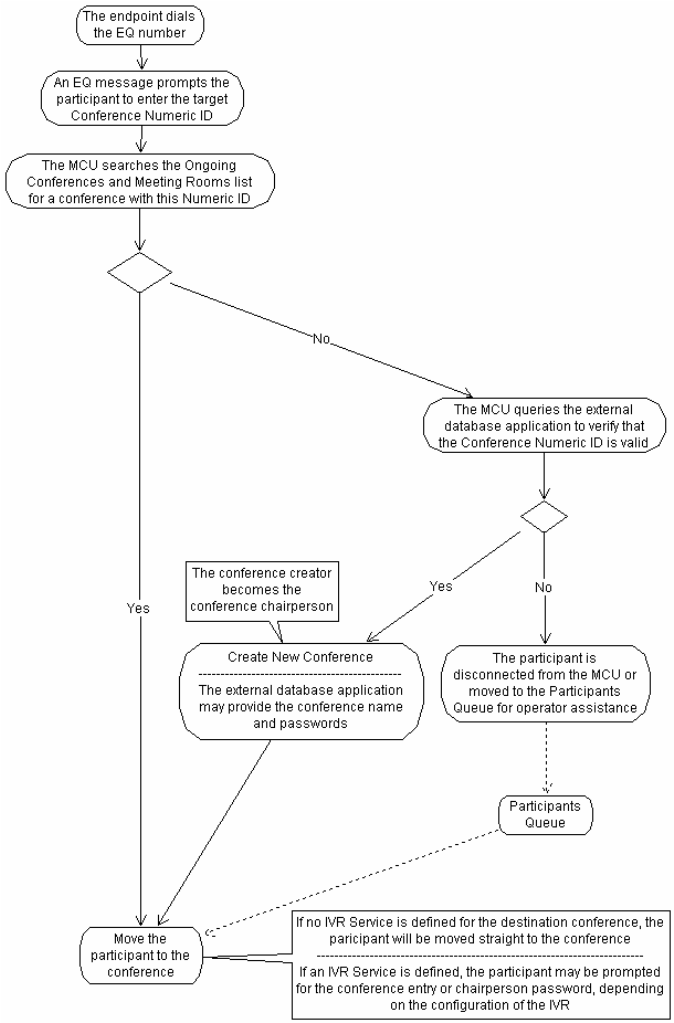
External Database Application Implementation Scenarios

Data Flow between the Entry Queue, the Conference IVR and the External Database Application



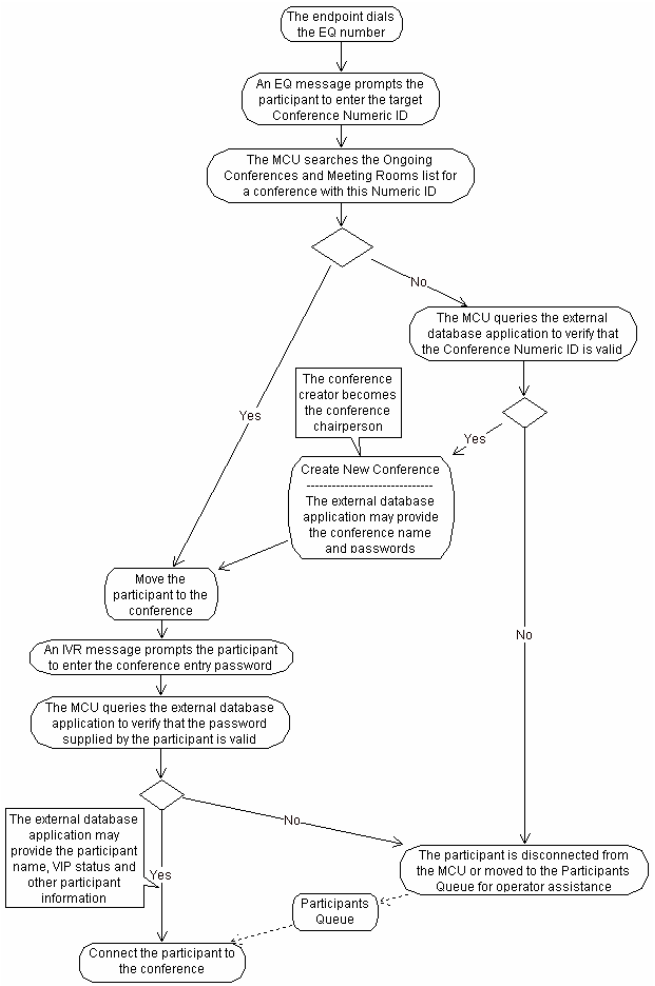
Scenario 1: Verifying the Participant Rights to Create a New Conference

This flow describes using an external database application only to verify the participant rights to create a new conference without password verification.



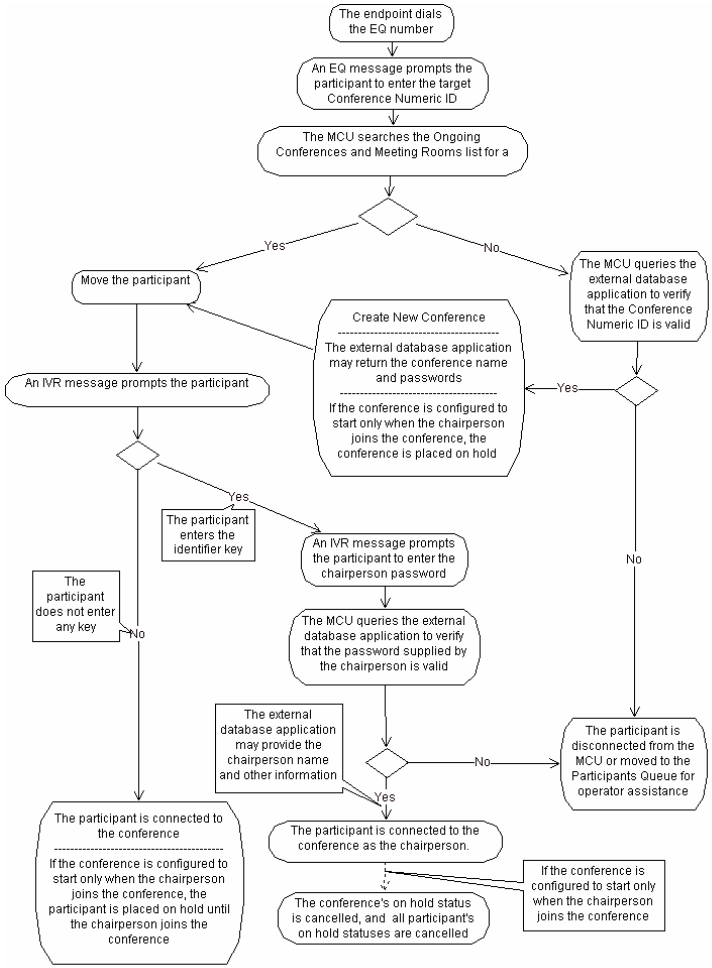
Scenario 2: Verifying the Participant Rights to Create a New Conference and Access the Conference

This flow describes using an external database application to verify the participant rights to create a new conference and in addition the participant's right to join the conference. The participant's right to join the conference can be performed independently, and it does not have to be as part of the Ad hoc conference creation flow.



Scenario 3: Verifying Participant Rights to Create a New Conference and to Join the Conference as Chairperson

This flow describes using an external database application to verify the participant rights to create a new conference and in addition the participant's right to join the conference as chairperson. The participant's right to join the conference as chairperson can be performed independently, and it does not have to be as part of the Ad hoc conference creation flow.



2

RMX Configuration for External Database Application Authorization

Before the MCU can use an external database application to verify a participant's rights to initiate a new conference or join an existing conference, or a user's authorization to log in to the MCU, you must perform the following configuration operations:

- Configure the MCU to use the external database application, by setting the appropriate flags in the system configuration. For more information, see "*Configuring the MCU to use an External Database Application*" on page **2-2**.
- To use an external database application to verify participant rights to create an Ad Hoc conference, configure the Entry Queue IVR Service assigned to the Ad Hoc-enabled Entry Queue accordingly. The configuration can be performed using the RMX Web Client or the RMX API. For more information, see "*Enabling External Database Validation for Starting New Ongoing Conferences*" on page **2-4**.
- To use an external database application to verify participant rights to join a conference using the conference password or chairperson password, configure the Conference IVR Service assigned to the Profile. The configuration can be performed using the RMX Web Client or the RMX API. For more information, see "*Enabling External Database Validation for Conferences Access*" on page **2-6**.



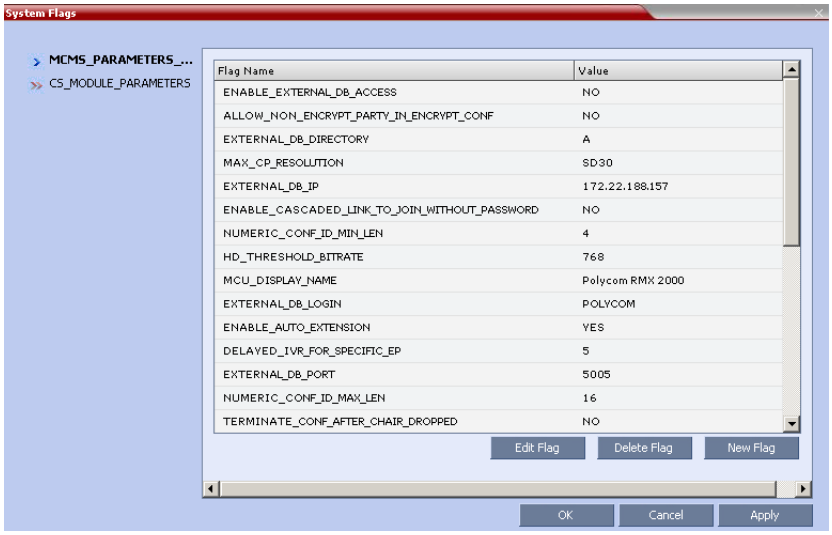
It is recommended that conference and/or chairperson passwords assigned to Ad Hoc conferences are provided by the external database application.

Configuring the MCU to use an External Database Application

Several flags must be set in the system configuration to define information related to the external database application.

To set the external database flags in the system configuration file:

1 In the RMX menu, click **Setup>System Configuration**.
The *System Flags* dialog box opens.



2 Modify the values of the following flags:

Table 2-1 *Flag Values for Accessing External Database Application*

Flag	Description and Value
ENABLE_EXTERNAL_DB_ACCESS	The flag that enables the use of the external database application.
EXTERNAL_DB_IP	The IP address of the external database application server. default IP: 0.0.0.0.

Table 2-1 *Flag Values for Accessing External Database Application (Continued)*

Flag	Description and Value
EXTERNAL_DB_ PORT	The port number used by the MCU to access the external application server. Default Port = 80. To use the WebCommander application as an external database application, you must specify 5005.
EXTERNAL_DB_ LOGIN	The user name defined in the external database application for the MCU. To use the WebCommander application as an external database application, the default user name is POLYCOM.
EXTERNAL_DB_ PASSWORD	The password associated with the user name defined for the MCU in the external database application. To use the WebCommander application as an external database application, the default password is POLYCOM.
EXTERNAL_DB_ DIRECTORY	The URL of the external database application.

For more information about flag setting, see *RMX Administrator's Guide, Chapter 11*.

3 Click **OK**.

4 Reset the MCU for flag changes to take effect.

Enabling External Database Validation for Starting New Ongoing Conferences

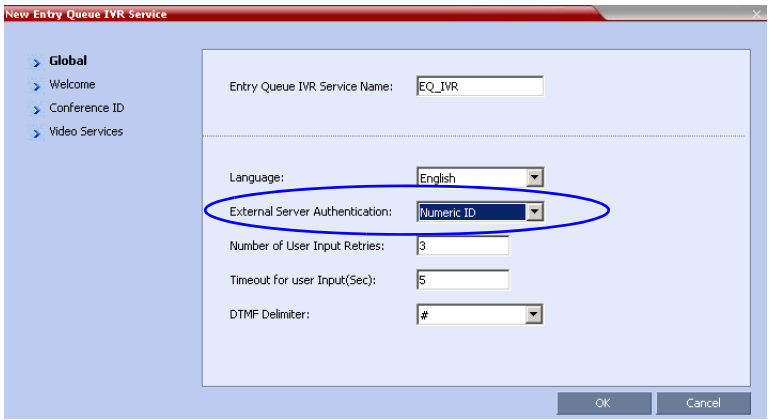
The Entry Queue IVR Service must be configured to perform the following operations:

- To use an external database application to verify participant rights to create a conference with a specified Conference ID. It is configured in the *Entry Queue IVR Service - Global* dialog box.
- To prompt the participant to enter a Conference ID. It is configured in the *Entry Queue IVR Service - Conference ID* dialog box

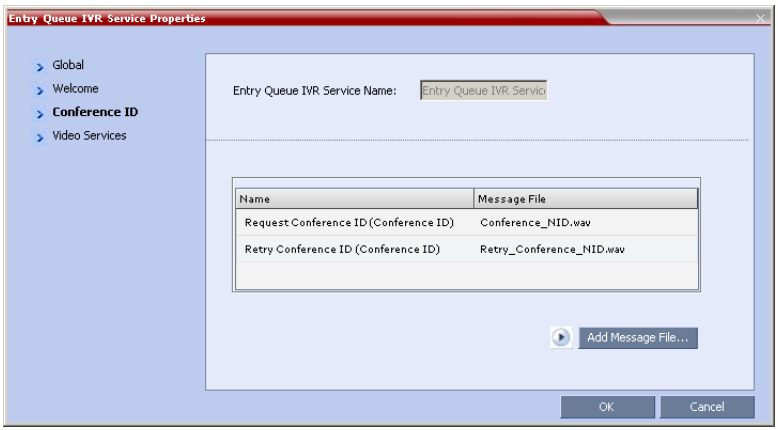
Using the RMX Web Client to Configure Entry Queue IVR Services

To define/modify the Entry Queue IVR Service:

- To validate the participant's right to start a new ongoing conference with an external database application:
  In the *Entry Queue IVR Service - Global* dialog box, set the *External Server Authentication* field to **Numeric ID**.



- In the *Entry Queue IVR Service - Conference ID* dialog box, select the appropriate audio files that prompt participant to enter or re-enter the Conference ID.



Using the RMX API to Configure Entry Queue IVR Services

All API functions in this procedure are in the ACCIVRMsg class.
If necessary, refer to the MGC API Software Developer's Guide for details of any function parameters not described in this procedure.

To configure an Entry Queue IVR Service using the RMX API:

1. Configure the Entry Queue IVR Service to use an external database application to verify that a conference can be created with the Conference ID specified by the participant, by applying the following API function to the Entry Queue IVR Service:

   ```
   void SetIVRExternalDB (ivr_external_db_numeric_id);
   ```

2. Specify the audio messages to be played by the Entry Queue IVR Service to prompt the participant to enter or re enter the Conference ID, by applying the following API function to the Entry Queue Message Service:

   ```
   void SetMsgFileName(const IVRFeature feature_opcode,
   const IVREvent event_opcode, const char* language_name,
   const char* file_name, int& status);
   ```

   — where the *feature_opcode* value is: **numeric_conf_id_feat**

— and the *event_opcode* value is:

ivr_event_get_numeric_id - for the message containing the initial request to enter the Conference ID

ivr_event_numeric_id_retry - for the message containing the request to re-enter the Conference Numeric ID

Enabling External Database Validation for Conferences Access

The Conference IVR Service can be configured to use an external database application to verify participant rights to join a conference in one of the following ways:

- By verifying the conference entry password.
- By verifying the chairperson password.



When the external database application is used, it can be configured to verify either the conference entry or the chairperson password, but not both.

The Conference IVR Service can be configured using the RMX Web Client or the RMX API.

Verifying the Conference Entry Password

The Conference IVR Service must be configured to:

- Verify the password or PIN code entered by the participant with an external database application. It is configured in the *Conference IVR Service - Global* dialog box.
- Prompt participants to enter the conference entry password.

Verifying the Chairperson Password

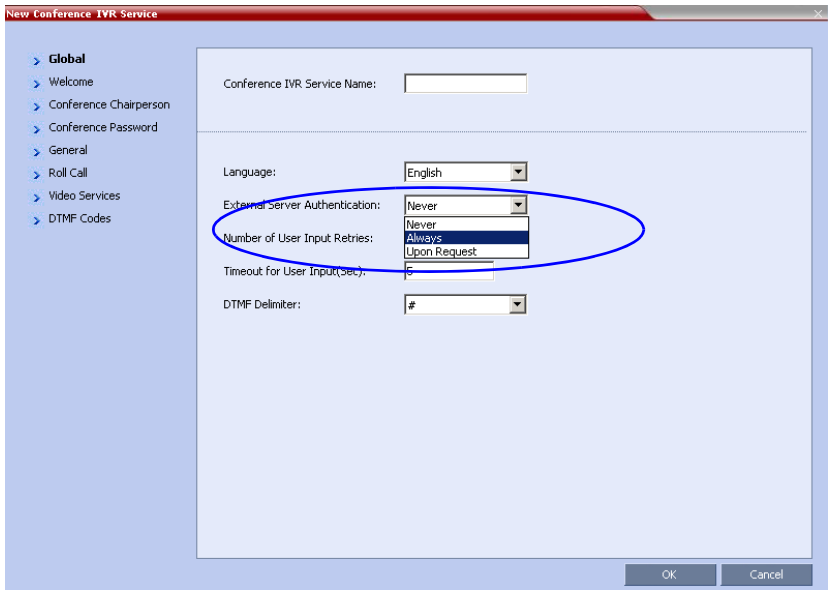
The Conference IVR Service must be configured to ask participants if they are the chairperson, and if they enter the chairperson identifier key, then to:

- Verify the password or PIN code entered by the participant with an external database application. It is configured in the *Conference IVR Service - Global* dialog box.
- Prompt the participant to enter the conference chairperson password.

Using the RMX Web Client to Configure Conference IVR Services

To define/modify the Conference IVR Service:

- To validate the participant's right to join the conference with an external database application:
  In the *Conference IVR Service - Global* dialog box, set the *External Server Authentication* field to:
  - **Always -** to validate the participant's right to join an ongoing conference for all participants
  - **Upon Request** - to validate the participant's right to join an ongoing conference as chairperson

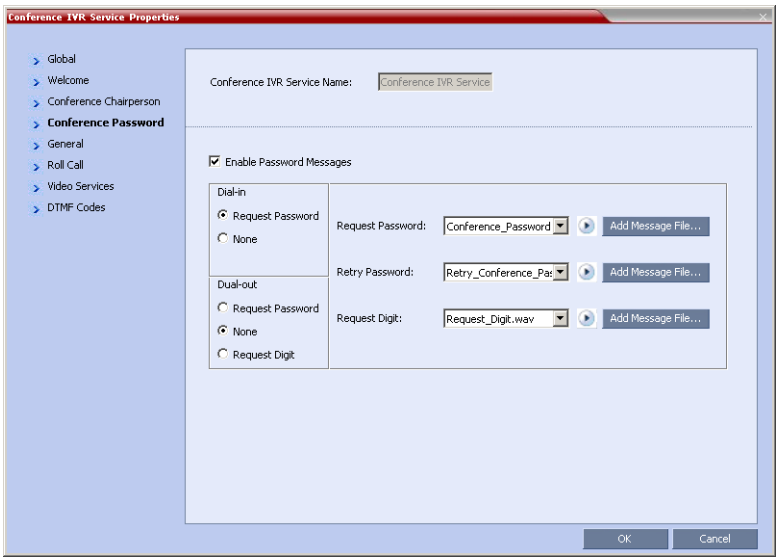


The external database application can be configured to check the participants PIN codes that they enter as either the conference password or chairperson password (depending on the IVR Service configuration).

- To prompt for the Conference Password and or Chairperson password, these options must be enabled in the Conference IVR Service and the appropriate audio files must be selected.

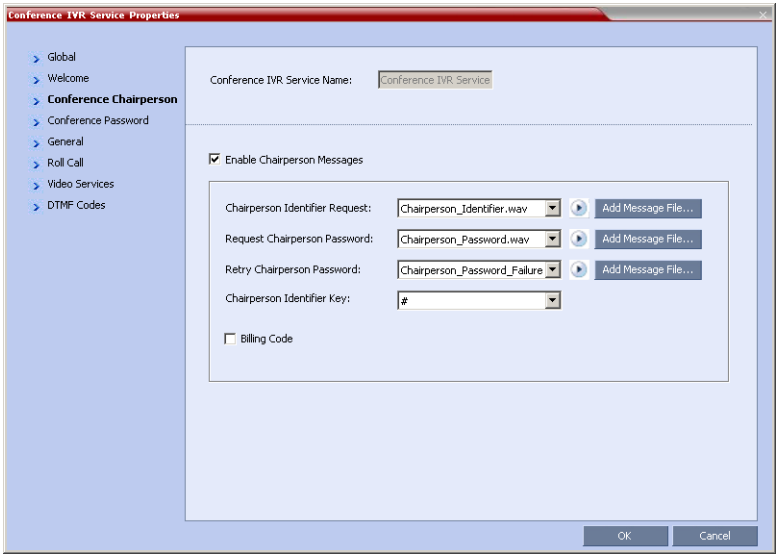
If the *External Server Authentication* option (step 1) is set to **Always**, conference entry password messages must be enabled, and this step is mandatory.
If the *External Server Authentication* option is set to **On Request**, enabling conference entry password messages is optional, and is only necessary if conference entry passwords are to be checked against the password specified in the conference definition.

— In the *Conference IVR Service - Conference Password* dialog box:



- Select the **Enable Password Messages** check box.
- In the *Dial-in* area, select the **Request Password** option.
- (Optional.) In the *Dial-out* area, select the **Request Password** option.
- In the *Request Password* list, select the audio file that prompts the participants for the password.
- In the *Retry Password* list, select the audio file that prompts the participants to re-enter the password, if they enter it incorrectly.

— In the *Conference IVR Service - Conference Chairperson* dialog box:



If the External Server Authentication option is set to **Upon Request**, then chairperson messages must be enabled, and this step is mandatory.

- Select the **Enable Chairperson Messages** check box.
- In the *Chairperson Identifier Request* list, select the audio file that requests users to enter the key that identifies them as the conference chairperson.
- In the *Request Chairperson Password* list, select the audio file that prompts the user for the chairperson password.
- In the *Chairperson Password failure* list, select the audio file that prompts the user to re-enter the chairperson password if they enter it incorrectly.
- In the *Chairperson Identifier Key* box, enter the key that will be used to identify the participant as a chairperson. Possible keys are: pound key (**#**) or star (*****).
- Ensure that the *Use Chairperson Password as Conf Password* check box is cleared, since it is not applicable when using an external database application to verify participant rights.

Using the RMX API to Configure Conference IVR Services

The configuration process using the RMX API has been divided into three procedures. The first procedure is mandatory, and the other two procedures are optional or mandatory depending whether the conference entry password or the chairperson password are to be verified with the external database application.



All API functions in the procedures are in the ACCIVRMsg class.

Procedure 1 - Configuring the Conference IVR Service to access an external database application to verify conference entry or chairperson passwords

This procedure is mandatory.

- Apply the following API function to the Conference IVR Service:
  `void SetIVRExternalDB (IVRExternalDB ExternalDB);`
  where the *ExternalDB* value depends which passwords are to be verified, as follows:
  - **ivr_external_db_participant_password** - to verify the conference password with an external database application. The verification process occurs for all participants attempting to join the conference. If required, the external database application can be written such that if a participant enters his/her PIN code when prompted to enter the conference entry password, then the PIN code will be verified in the external database.
  - **ivr_external_db_chair_password** - to verify only the chairperson password with an external database application. The verification process occurs only if the participant enters the chairperson identifier key (pound or star). In this case, all other participants are connected to the conference as standard participants. If required, the external database application can be written such that if a participant enters his/her PIN code when prompted to enter the chairperson password, then the PIN code will be verified in the external database.

Procedure 2 - Configuring the Conference IVR Service to prompt the participant to enter the conference entry password

If the ExternalDB value is set to ***ivr_external_db_participant_password*** (in the first procedure), then conference entry password messages must be enabled, and this step is mandatory.

If the ExternalDB value is set to is set to ***ivr_external_db_chair_password***, enabling conference password messages is optional, and is only necessary if conference entry passwords are to be checked against the password specified in the conference definition.

1. Configure the Conference IVR Service to prompt dial-in participants to enter the conference password, by applying the following API function to the Conference IVR Service:

   ```
   void SetDialInPasswordRequest (pwd_request_password);
   ```

2. Optional. Configure the Conference IVR Service to prompt dial-out participants to enter the conference password, by applying the following API function to the Conference IVR Service:

   ```
   void SetDialOutPasswordRequest(pwd_request_password);
   ```

3. Specify the messages to be played by the Conference IVR Service to prompt the participant to enter or re-enter the conference password, by applying the following API function to the Conference IVR Service:

   ```
   void SetMsgFileName(const IVRFeature feature_opcode,
   const IVREvent event_opcode, const char* language_name,
   const char* file_name, int& status);
   ```

   where the *feature_opcode* value is: **conf_pass_feat**

   and the *event_opcode* value is:

   - **get_conf_pass_event**- for the message containing the initial request to enter the conference password
   - **pass_ret_event**- for the message containing the request to re-enter the conference password

Procedure 3 - Configuring the Conference IVR Service to prompt the participant for the chairperson identifier key and chairperson password

If the ExternalDB value is set to is set to ***ivr_external_db_chair_password*** (in the first procedure), then chairperson messages must be enabled, and this step is mandatory.

If the ExternalDB value is set to ***ivr_external_db_participant_password***, then the MCU only checks one password with the external database application (usually the conference password) and therefore enabling chairperson messages is optional, and is only necessary if chairperson passwords are to be checked against the password specified in the conference definition.

1. Specify the audio message to be played by the Conference IVR Service prompting the participant to enter the chairperson identifier key, by applying the following API function to the Conference IVR Service:

   ```
   void SetMsgFileName(const IVRFeature feature_opcode,
   const IVREvent event_opcode, const char* language_name,
   const char* file_name, int& status);
   ```

   where the *feature_opcode* value is: **leader_feat**

   and the *event_opcode* value is: **get_leader_identifier_event**

2. Specify the messages to be played by the Conference IVR Service to prompt the participant to enter or re-enter the chairperson password, by applying the following API function to the Conference IVR Service:

   ```
   void SetMsgFileName(const IVRFeature feature_opcode,
   const IVREvent event_opcode, const char* language_name,
   const char* file_name, int& status);
   ```

   where the *feature_opcode* value is: **leader_feat**

   and the *event_opcode* value is:

   — **get_leader_password_event** - for the message containing the initial request to enter the chairperson password

   — l**eader_password_retry_event** - for the message containing the request to re-enter chairperson password

3 Specify the key that will be used to identify the participant as a chairperson, by applying the following API function to the Conference IVR Service:

```
void SetLeaderIdentifier(const char*
szLeader_identifier);
```

where the *szLeader_identifier* value is: **#** (pound key) or ***** (star)

4 Simplify the conference login process for chairpersons, enabling the them to enter only their chairperson passwords without entering the conference entry passwords, by applying the following API function to the Conference IVR Service:

```
void SetConfPwAsLeaderPw(TRUE);
```

3

Working with the External Database API

About the XML Requests and Responses

The MCU communicates with the external database application using the External Database API, which consists of XML requests and responses. There are four types of XML requests and corresponding responses: Create Conference, Add Participant, Login and Keep Alive.

Each XML request from the MCU to the external database application starts with a LOGIN section containing the IP address of the MCU sending the request, and a user name and password to identify the MCU to the external database application.

The **Create Conference** request verifies that a specific participant has the necessary rights to start a conference with a specific Conference ID. This request is only relevant to Ad Hoc conferences. The Create Conference response validates the Create Conference request, and can return conference information such as the conference billing code, and the conference entry and chairperson passwords.

The **Add Participant** request verifies that a specific participant has the necessary rights to join a conference with a specific Conference ID, and can also verify the conference entry or chairperson password, or the participant PIN code. This request is relevant to both Ad Hoc and pre-defined conferences. The Add Participant response validates the Add Participant request, and can return information such as whether or not the participant is a conference chairperson.

The **Login** request verifies that a specific user is authorized to log into the MCU with a specific password. The Login response validates the Login request, and can return the user's authorization level.

The **Keep Alive** request is used by the MCU as a client to perform a keep-alive check with the external server every 2 minutes. If the connection to the external server is closed, the Keep Alive request will re-open it. If a Create Conference, Add Participant or Login request are sent to the external server when the connection is closed, these requests will re-open the connection.



By default Apache closes a connection if no request is received in 15 seconds. As a result the MCU has to keep re-opening the connection. To prevent this it is recommended to change the Apache timeout configuration to more than two minutes, for example "KeepAliveTimeout 140".
The default IIS timeout period is 900 seconds, and therefore does not need modification.

If the server does not respond to the Keep Alive request, the MCU will go into MAJOR error status with the status 'EXTERNAL DB APPLICATION DOES NOT RESPOND'. This error will be written in the MCU log file. The MCU will continue performing keep-alive checks with the server every 2 minutes, and will stay in MAJOR error status until the external server responds. During this time any call that requires validation from the external server, for example, a request for validation to create a new conference, will be rejected. Any response from the external server regarding prior requests will not be processed by the MCU as long as the MCU remains in MAJOR error state.

This guide only describes the External Database API requests and responses that are applicable to standard MCUs.

The Syntax of the XML Requests and Responses



The term mandatory in the description of an element in the request XML indicates that the MCU always sends this element. Other elements are optional.

The Create Conference XML

The Create Conference Request

The MCU sends the following XML request to the external database application:



The element values will be replaced by the values relevant to the specific Create Conference request.

```
<REQUEST_CONF_DETAILS>
    <LOGIN>
        <MCU_IP>127.0.0.1</MCU_IP>
        <USER_NAME>POLYCOM</USER_NAME>
        <PASSWORD>POLYCOM</PASSWORD>
    </LOGIN>
    <TOKEN>5</TOKEN>
    <ACTION>
        <CREATE>
            <NUMERIC_ID>123456</NUMERIC_ID>
            <ACTUAL_PARTY_PHONES>
                <PHONE1>111</PHONE1>
                <PHONE2>222</PHONE2>
            </ACTUAL_PARTY_PHONES>
        </CREATE>
    </ACTION>
</REQUEST_CONF_DETAILS>
```

The element values in the Create Conference request are as follows:

***Table 3-1 MCU Login Information** (<LOGIN>)*

Element	Description
MCU_IP	The IP address of the MCU sending the request.
USER_NAME	The user name to identify the MCU to the external database application. The default value is POLYCOM.
PASSWORD	The password to identify the MCU to the external database application. The default value is POLYCOM
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

***Table 3-2 Conference Creation Information** (<ACTION><CREATE>)*

Element	Description
NUMERIC_ID	Mandatory. The Conference Numeric ID of the destination conference. This ID is used to validate the participant request to start a new conference.
<ACTUAL_PARTY_ PHONES>	The participant alias. Consists of the <PHONE1> and <PHONE2> elements as follows: **PHONE1** For H.323 participants, the first E.164 number, or the first H.323 ID if no E.164 number was delivered. For ISDN participants: The calling participant CLI number. (Available only if this information has been passed to the MCU.) **PHONE2** Not applicable to IP participants. For ISDN participants: The second calling participant CLI number, where applicable.

The Create Conference Response

The external database application must send the following XML response to the MCU:



The external database application must replace the element values with the values relevant to the specific Create Conference request.

```
<CONFIRM_CONF_DETAILS>
    <RETURN_STATUS>
        <ID>0</ID>
        <DESCRIPTION>OK</DESCRIPTION>
    </RETURN_STATUS>
    <TOKEN>5</TOKEN>
    <ACTION>
        <CREATE>
            <NAME>Conf_1</NAME>
            <MAX_PARTIES>10</MAX_PARTIES>
            <MIN_NUM_OF_PARTIES>3</MIN_NUM_OF_PARTIES>
            <PASSWORD>111111</PASSWORD>
            <ENTRY_PASSWORD>222222</ENTRY_PASSWORD>
            <BILLING_DATA>POLYCOM</BILLING_DATA>
            <OWNER>POLYCOM</OWNER>
            <CONTACT_INFO_LIST>
                <CONTACT_INFO>Remark1</CONTACT_INFO>
                <CONTACT_INFO>Remark2</CONTACT_INFO>
                <CONTACT_INFO>Remark3</CONTACT_INFO>
            </CONTACT_INFO_LIST>
            <DISPLAY_NAME>Conf_1</DISPLAY_NAME>
        </CREATE>
    </ACTION>
</CONFIRM_CONF_DETAILS>
```

The element values in the Create Conference response are as follows:

***Table 3-3 Return Status Information** (<RETURN_STATUS>)*

Element	Description
ID	Mandatory. The return status code. For valid values see Table 3-5 on page **3-8**.
DESCRIPTION	The return status description. You can provide your own return status description of up to 50 characters, or you can use the status descriptions provided in Table 3-5 on page **3-8**.
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

***Table 3-4 Conference Creation Information** (<ACTION><CREATE>)*

Element	Description
NAME	The name to be assigned to the conference, for example, 100234_PeterD (ID_Creator_Name).
MAX_PARTIES	The maximum number of participants allowed in the conference. This value overrides the maximum participants value in the Profile assigned to the Entry Queue, and enables a different value to be used for each conference creator. **Note:** The MAX_PARTIES element is optional. If the external database application does not return a value for this element, then the maximum participants value is taken from the Profile assigned to the Entry Queue. When the WebCommander is used as the external database application, the MAX_PARTIES element is not returned, and therefore the maximum parties value is taken from the Profile.
MIN_NUM_OF_ PARTIES	The minimum number of participants for the conference. This value enables a different value to be used for each conference creator.

***Table 3-4 Conference Creation Information** (<ACTION><CREATE>)*

Element	Description
MIN_NUM_OF_ PARTIES (cont.)	**Note:** The MIN_NUM_OF_PARTIES element is optional. if the external database application does not return a value for this element, then the minimum participants value is 0 (the default system setting). When the WebCommander is used as the external database application, the MIN_NUM_OF_PARTIES element is not returned. If insufficient resources are available on the MCU for the specified number of participants, the conference will not be set up.
PASSWORD	The chairperson password for the conference. If a value is specified for this field, and the Conference IVR Service used by the destination conference is configured to verify with the external database application before adding a participant to the conference, then the <LEADER> element will be set to **true** in the *Add Participant* request. **Note:** The PASSWORD element is optional, but if chairperson passwords are to be used for conferences, the chairperson passwords should be provided by the external database application, otherwise a random password will be assigned by the MCU when creating the conference.
ENTRY_ PASSWORD	The conference entry password for the conference. If a value is specified for this field, and the IVR Service used by the destination conference is configured to verify with the external database application before adding a participant to the conference, then the <GUEST> element will be set to true in the Add Participant request, and the participant is added as an undefined standard participant. **Note:** This value overrides any conference password assigned by the MCU. The ENTRY_PASSWORD element is optional, but if conference passwords are to be used for conferences, the conference passwords should be provided by the external database application, otherwise a random password will be assigned by the MCU when creating the conference.

Table 3-4 ***Conference Creation Information*** *(<ACTION><CREATE>)*

Element	Description
BILLING_DATA	The billing code for the conference. The billing code value is saved to the CDR file and can be retrieved later. **Note:** This value overrides any billing code specified in the Profile assigned to the Entry Queue.
OWNER	The name of the conference owner. This value will be added to the conference remarks field and saved in the CDR file.
<CONTACT_ INFO_LIST>	Text to be put in the Info fields for the conference. The Info fields for the conference enable general information to be entered for the conference, such as the company name, the contact person name, the contact person's e-mail or telephone number, or any other information required. The text in the Info fields is saved to the CDR file and can be retrieved later. Consists of three <CONTACT_INFO> elements as follows: **CONTACT_INFO** The text to be placed in the first Info (Info1) field. **CONTACT_INFO** The text to be placed in second Info (Info2) field. **CONTACT_INFO** The text to be placed in third Info (Info3) field.
<DISPLAY_ NAME>	The conference display name.

Table 3-5 *Permitted Status Values for the Create Conference Response*

ID	Description	Explanation
0	OK	The MCU can create the conference.
1	Invalid MCU user name and/or password	The user name and/or password used by the MCU do not match the user names/passwords defined in the external database application.

Table 3-5 *Permitted Status Values for the Create Conference Response*

ID	Description	Explanation
3	Invalid NID	The requested Conference Numeric ID is not registered in the external database application, and therefore the creation of a new conference is not permitted.
4	Invalid CLI	The participant with the CLI number sent from the MCU is not permitted to create a new conference.
5	Invalid CLI/NID	The participant with the CLI number sent from the MCU is not permitted to create a new conference with the specified Numeric ID.
6	Internal Error	An error has occurred in the external database application.

The Add Participant XML

The Add Participant Request

The MCU sends the following XML request to the external database application:



The element values will be replaced by the values relevant to the specific Add Participant request.

```
<REQUEST_PARTY_DETAILS>
    <LOGIN>
        <MCU_IP>127.0.0.1</MCU_IP>
        <USER_NAME>POLYCOM</USER_NAME>
        <PASSWORD>POLYCOM</PASSWORD>
    </LOGIN>
    <TOKEN>5</TOKEN>
    <ACTION>
        <ADD>
            <NUMERIC_ID>123456</NUMERIC_ID>
            <PASSWORD>111111</PASSWORD>
            <LEADER>true</LEADER>
            <GUEST>false</GUEST>
            <ACTUAL_PARTY_PHONES>
                <PHONE1>111</PHONE1>
                <PHONE2>222</PHONE2>
            </ACTUAL_PARTY_PHONES>
            <PARTY_ID>1</PARTY_ID>
        </ADD>
    </ACTION>
</REQUEST_PARTY_DETAILS>
```

The element values in the Add Participant request are as follows:

***Table 3-6 MCU Login Information** (<LOGIN>)*

Element	Description
MCU_IP	The IP address of the MCU sending the request.
USER_NAME	The user name to identify the MCU to the external database application. The default value is POLYCOM.
PASSWORD	The password to identify the MCU to the external database application. The default value is POLYCOM
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

***Table 3-7 Participant Addition Information** (<ACTION>< ADD >)*

Element	Description
NUMERIC_ID	Mandatory. The ID of the conference that the participant wishes to join
PASSWORD	Mandatory. The password or PIN code entered by the participant.
LEADER	Indicates whether or not the external database application should verify that the password entered by the participant is a valid chairperson password for the participant for the specified conference, as follows: **true** Verifies that the specified password is a valid chairperson password for the conference. **false** Does not verify that the specified password is a valid chairperson password for the conference. Default. **Note:** If the external database application returned a chairperson password in the PASSWORD element in the Create Conference response, then the LEADER element will be set to **true**.

Table 3-7 ***Participant Addition Information*** *(<ACTION>< ADD >) (Continued)*

Element	Description
GUEST	Indicates whether or not the external database application should verify that the password entered by the participant is a valid conference entry password for the participant for the specified conference, as follows: **true** Verifies that the password is a valid entry password for the conference. **false** Does not verify that the password is a valid entry password for the conference. Default. **Note:** If the external database application returned an entry password in the ENTRY_PASSWORD element in the Create Conference response, then the GUEST element will be set to **true**.
<ACTUAL_PARTY_PHONES>	The participant alias. Consists of the <PHONE1> and <PHONE2> elements as follows: **PHONE1** For H.323 participants, the first E.164 number, or the first H.323 ID if no E.164 number was delivered. For ISDN participants: The calling participant CLI number. (Available only if this information has been passed to the MCU.) **PHONE2** Not applicable to IP participants. For ISDN participants: The second calling participant CLI number, where applicable.
<PARTY_ID>	The participant ID

The Add Participant Response

The external database application must send the following XML response to the MCU:



The external database application must replace the element values with the values relevant to the specific Add Participant request.

```
<CONFIRM_PARTY_DETAILS>
    <RETURN_STATUS>
        <ID>0</ID>
        <DESCRIPTION>OK</DESCRIPTION>
    </RETURN_STATUS>
    <TOKEN>5</TOKEN>
    <ACTION>
        <ADD>
            <LEADER>true</LEADER>
            <NAME>Party_1</NAME>
            <VIP>true</VIP>
            <CONTACT_INFO_LIST>
                <CONTACT_INFO>Remark1</CONTACT_INFO>
                <CONTACT_INFO>Remark2</CONTACT_INFO>
                <CONTACT_INFO>Remark3</CONTACT_INFO>
                <CONTACT_INFO>Remark4</CONTACT_INFO>
            </CONTACT_INFO_LIST>
        </ADD>
    </ACTION>
</CONFIRM_PARTY_DETAILS>
```

The element values in the Add Participant response are as follows:

***Table 3-8 Return Status Information** (<RETURN_STATUS>)*

Element	Description
ID	Mandatory. The return status code. For valid values see Table 3-10 on page **3-15**.

Table 3-8 ***Return Status Information*** *(<RETURN_STATUS>) (Continued)*

Element	Description
DESCRIPTION	The return status description. You can provide your own return status description of up to 50 characters, or you can use the status descriptions provided in Table 3-10 on page **3-15**.
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

Table 3-9 ***Participant Addition Information*** *(<ACTION><ADD>)*

Element	Description
LEADER	Mandatory. Indicates whether or not the participant is a conference chairperson, as follows: **true** The participant is a conference chairperson. **false** The participant is not a conference chairperson. Default. **Note:** This value overrides the status assigned to the participant in the Entry Queue.
NAME	The name to be assigned to the participant. This name will be used as the visual name for the participant endpoint.
VIP	Indicates whether the participant is a VIP participant or a standard participant, as follows: **true** The participant is a VIP participant. **false** The participant is a standard participant. Default. **Note:** *This element is currently not supported in the RMX.*

Table 3-9 ***Participant Addition Information*** *(<ACTION><ADD>) (Continued)*

Element	Description
<CONTACT_ INFO_LIST>	Text to be inserted in the Info fields for the participant. The Info fields for the participant enable general information to be entered for the participant, such as the participant's e-mail address or any other information required. The text in Info fields is saved to the CDR file and can be retrieved later.
<CONTACT_ INFO_LIST> (cont.)	Consists of four <CONTACT_INFO> elements as follows: **CONTACT_INFO** The text to be added in the first Info field (Info1). **CONTACT_INFO** The text to be added in the second Info field (Info2). **CONTACT_INFO** The text to be added in the third Info field (Info3). **CONTACT_INFO** The text to be added in the fourth Info field (Info4).

Table 3-10 *Permitted Status Values for the Add Participant Response*

ID	Description	Explanation
0	OK	The MCU can permit the participant to join the conference.
1	Invalid MCU user name and/or password	The user name and/or password used by the MCU do not match the user names and/or passwords defined in the external database application.
3	Invalid ID / Password	The ID and/or password entered by the participant are not registered in the external database application, and therefore the participant is not permitted to join the conference.
4	Invalid CLI	The participant with the CLI number sent from the MCU is not permitted to join the conference.
5	Invalid CLI/NID	The combination of the CLI number sent from the MCU and the requested Numeric ID is not permitted by the external database application.

Table 3-10 *Permitted Status Values for the Add Participant Response*

ID	Description	Explanation
6	Internal Error	An error occurred in the external database application.

The Login XML

The Login Request

The MCU sends the following XML request to the external database application:



The element values will be replaced by the values relevant to the specific Login request.

```
<REQUEST_USER_DETAILS>
    <LOGIN>
        <MCU_IP>127.0.0.1</MCU_IP>
        <USER_NAME>POLYCOM</USER_NAME>
        <PASSWORD>POLYCOM</PASSWORD>
    </LOGIN>
    <TOKEN>5</TOKEN>
    <ACTION>
        <AUTHENTICATE>
            <USER_NAME>JUDITH</USER_NAME>
            <PASSWORD>D4MP45</PASSWORD>
            <STATION_NAME>F3-JUDITHS</STATION_NAME>
            <STATION_IP>172.22.164.7</STATION_IP>
        </AUTHENTICATE>
    </ACTION>
</REQUEST_USER_DETAILS>
```

The element values in the Login request are as follows:

***Table 3-11 MCU Login Information** (<LOGIN>)*

Element	Description
MCU_IP	The IP address of the MCU sending the request.
USER_NAME	The user name to identify the MCU to the external database application. The default value is POLYCOM.
PASSWORD	The password to identify the MCU to the external database application. The default value is POLYCOM
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

***Table 3-12 User Login Information** (<ACTION><AUTHENTICATE>)*

Element	Description
USER_NAME	Mandatory. The login name entered by the User.
PASSWORD	Mandatory. The password entered by the User.
STATION_NAME	The name of the workstation from which the connection is being made.
STATION_IP	The IP address of the workstation from which the connection is being made.

The Login Response

The external database application must send the following XML response to the MCU:



The external database application must replace the element values with the values relevant to the specific Login request.

```
<CONFIRM_USER_DETAILS>
    <RETURN_STATUS>
        <ID>0</ID>
        <DESCRIPTION>OK</DESCRIPTION>
    </RETURN_STATUS>
    <TOKEN>5</TOKEN>
    <ACTION>
            <AUTHENTICATE>
                <AUTHORIZATION_GROUP>operator</
                AUTHORIZATION_GROUP>
            </AUTHENTICATE>
    </ACTION>
</CONFIRM_USER_DETAILS>
```

The element values in the Login response are as follows:

***Table 3-13 Return Status Information** (<RETURN_STATUS>)*

Element	Description
ID	Mandatory. The return status code. For valid values see Table 3-10 on page **3-15**.
DESCRIPTION	The return status description. You can provide your own return status description of up to 50 characters, or you can use the status descriptions provided in Table 3-10 on page **3-15**.
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

***Table 3-14 Login Information** (<ACTION><AUTHENTICATE>)*

Element	Description
AUTHORIZATION_GROUP	Mandatory. The user authorization level, as follows: **administrator** The user is an administrator. Default. Users of this type can define and manage conferences, Meeting Rooms, and participants. In addition, these users can change the MCU configuration, including defining and deleting other users, and defining Network Services. **operator** The user is an Operator. Users of this type can define and manage conferences, Meeting Rooms, and participants. In addition, these users can view certain MCU configurations, but they cannot change them. **attendant** The user is a chairperson. Users of this type can define and manage ongoing conferences and participants. These users cannot view the MCU configuration.

***Table 3-15** Permitted Status Values for the Login Response*

ID	Description	Explanation
0	OK	The MCU can permit the user to log in to the MCU.
1	Invalid MCU user name and/or password	The user name and/or password used by the MCU do not match the user names and/or passwords defined in the external database application.
8	Invalid user name and/or password	The user name and/or password specified by the user do not match the user names and/or passwords defined in the external database application.

The Keep Alive XML

The Keep Alive Request

The MCU sends the following XML request to the external database application:



The element values will be replaced by the values relevant to specific MCU and external database application.

```
<REQUEST_GENERAL>
    <LOGIN>
        <MCU_IP>127.0.0.1</MCU_IP>
        <USER_NAME>POLYCOM</USER_NAME>
        <PASSWORD>POLYCOM</PASSWORD>
    </LOGIN>
    <TOKEN>5</TOKEN>
    <ACTION>
        <KEEP_ALIVE/>
    </ACTION>
</REQUEST_ GENERAL>
```

The element values in the Keep Alive request are as follows:

***Table 3-16 MCU Login Information** (<LOGIN>)*

Element	Description
MCU_IP	The IP address of the MCU sending the request.
USER_NAME	The user name to identify the MCU to the external database application. The default value is POLYCOM.
PASSWORD	The password to identify the MCU to the external database application. The default value is POLYCOM
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

The Keep Alive Response

The external database application must send the following XML response to the MCU:



The external database application must replace the element values with the values that are applicable at the time the Keep Alive request is received.

```
<CONFIRM_GENERAL>
    <RETURN_STATUS>
            <ID>0</ID>
            <DESCRIPTION>OK</DESCRIPTION>
    </RETURN_STATUS>
    <TOKEN>5</TOKEN>
    <ACTION>
            <KEEP_ALIVE/>
    </ACTION>
</CONFIRM_GENERAL>
```

The element values in the Keep Alive response are as follows:
:

***Table 3-17 Return Status Information** (<RETURN_STATUS>)*

Element	Description
ID	Mandatory. The return status code. For valid values see Table 3-18 on page **3-22**.
DESCRIPTION	The return status description. You can provide your own return status description of up to 50 characters, or you can use the status descriptions provided in Table 3-18 on page **3-22**.
TOKEN	An internal value sent by the MCU that must be returned by the external database application as is.

Table 3-18 Permitted Status Values for the Login Response

ID	Description	Explanation
0	OK	The application server is alive.
1	Invalid MCU user name and/or password	The user name and/or password used by the MCU do not match the user names and/or passwords defined in the external database application.
2	Internal Error	An error occurred in the external database application.

4

Using the Sample Scripts

About the Sample Scripts

The sample scripts can be used to communicate with a directory type database, as an alternative to writing an application which receives the XML requests, and returns responses in the required XML format. The user has to provide the mapping between the directory attributes and the XML elements, using the supplied scripts.



- The sample scripts should work with any directory, but have only been tested with iPlanet Version 5.1 and MS Active Directory.
- The current version of the scripts only supports anonymous access to the directory.
- For Login requests, the scripts only check domain users.

System Requirements

- Windows 2000®, Windows 2003® or Windows XP® (For Windows 2000 you must have SP3 or higher.)
- IIS

Installation and Configuration Procedures

Before you can use the sample scripts, you must perform the following installation and configuration operations:

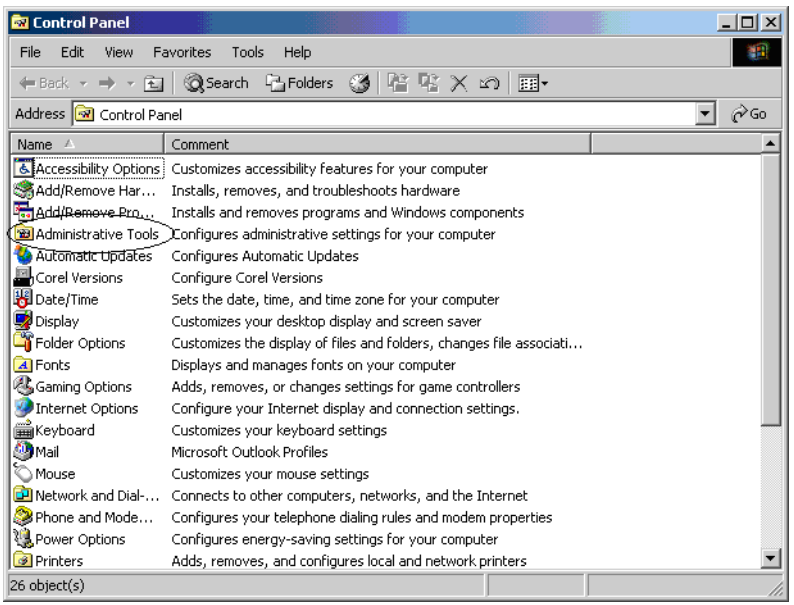
1. Copy the supplied script files to a directory on your server.
2. Create a virtual directory in IIS. For more information, see "*Creating a Virtual Directory in IIS*" on page **4-3**.
3. Restrict access to the directory configuration files. For more information, see "*Defining Security Properties for the Directory Configuration Files*" on page **4-8**.
4. Define the mapping between the directory fields and the XML fields, by configuring the Directory Configuration file. For more information, see "*Configuring the Directory Configuration File*" on page **4-16**.
5. Check the mapping using the supplied simulator. For more information, see "*Testing the Directory Configuration File using the External Database Simulator*" on page **4-22**.

Creating a Virtual Directory in IIS

To use the sample scripts. create a virtual directory in IIS and map it to the directory to which you copied the script files.

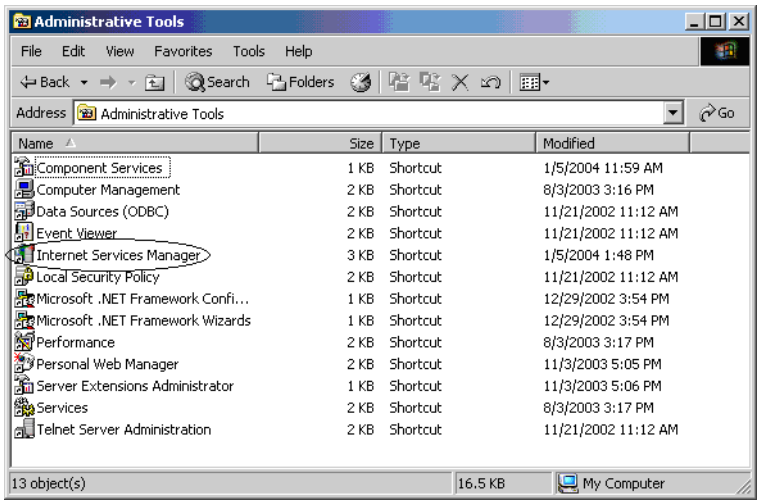
To create a virtual directory for the directory containing the script files:

1 Open the Windows Control Panel and double-click the **Administrative Tools** icon.



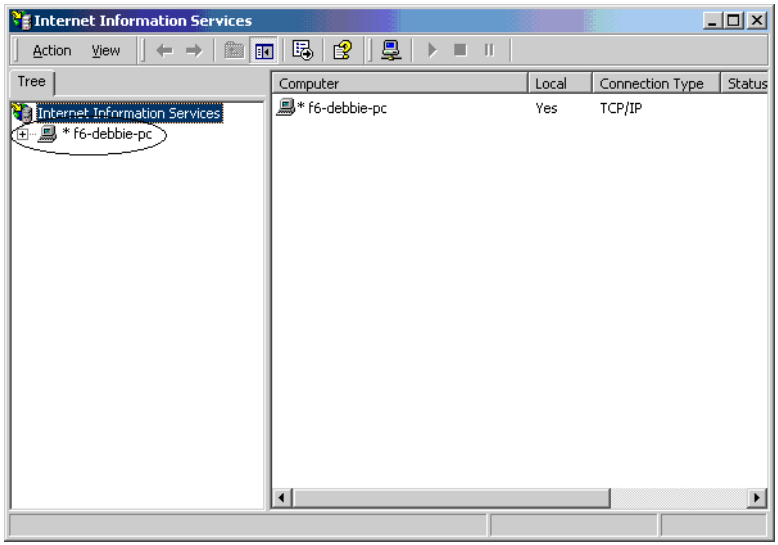
The *Administrative Tools* window opens.

2 Double-click the **Internet Services Manager** icon.

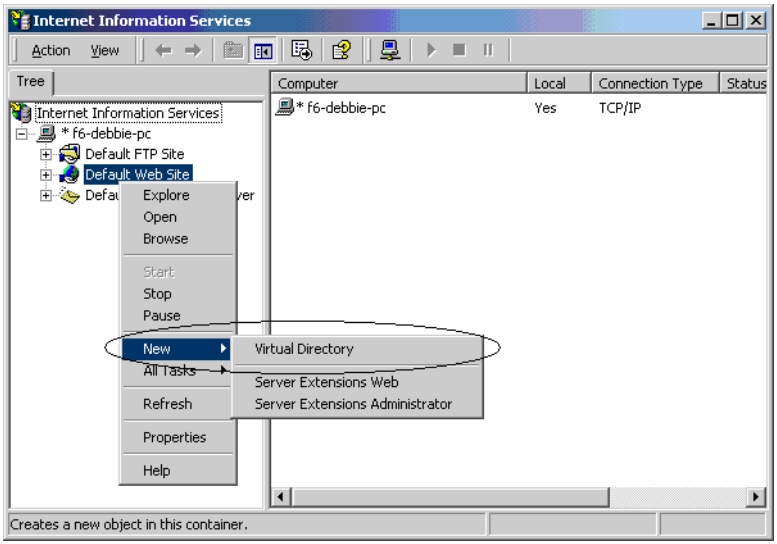


The *Internet Information Services* window opens.

3 Expand the Server list.



4 Right-click the **Default Web Site** icon, click **New**, and then click **Virtual Directory**.

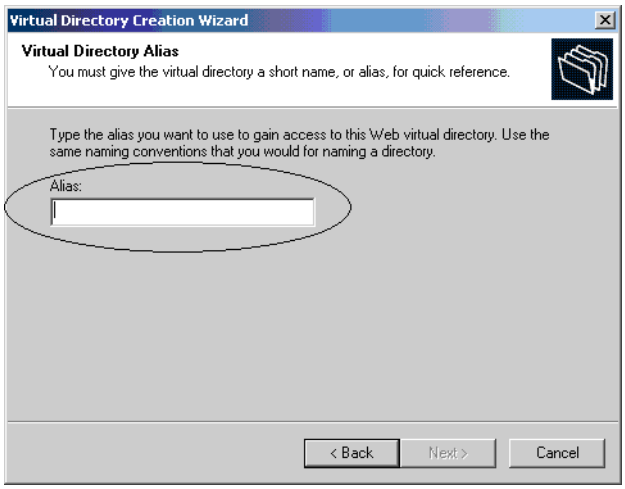


The *Virtual Directory Creation Wizard* dialog box opens.

5 Click **Next**.

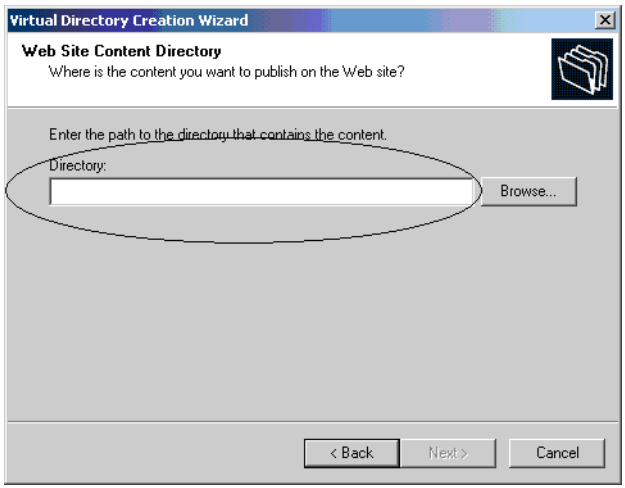
The *Virtual Directory Alias* dialog box opens.

6 In the *Alias* box, enter an alias name for the virtual directory, and then click **Next**.



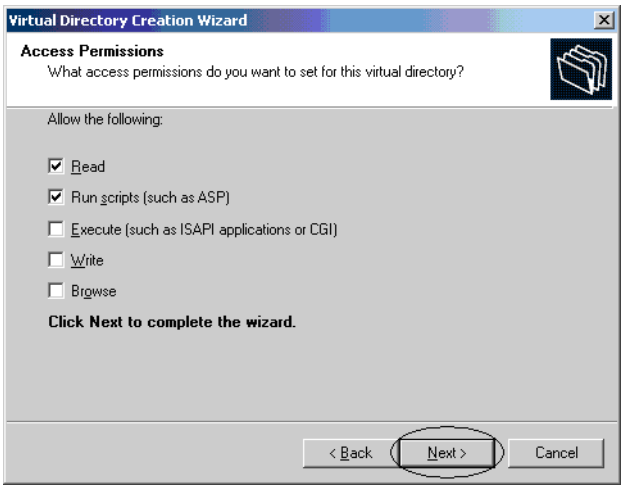
The *Web Site Content Directory* dialog box opens.

7 In the *Directory* box, enter the path to the directory containing the sample script files, or click the **Browse** button and select the Directory. Click **Next**.



The *Access Permissions* dialog box opens.

8 Click **Next.**



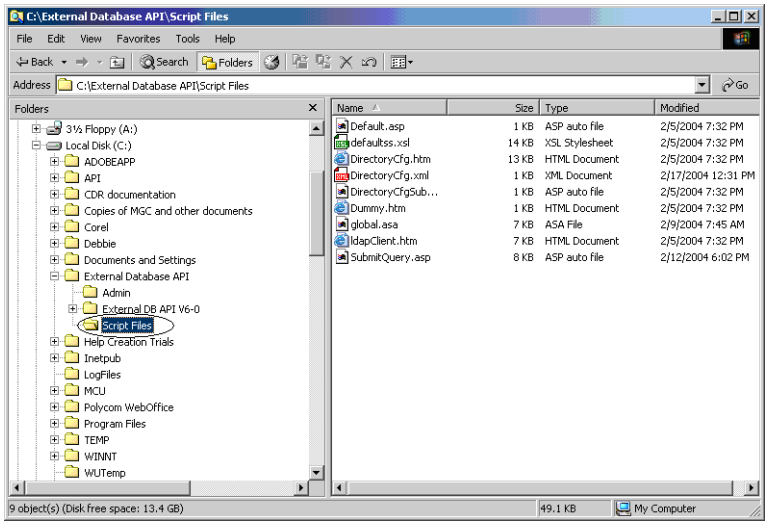
9 Click **Finish** to complete the procedure.

Defining Security Properties for the Directory Configuration Files

To avoid unauthorized access to the directory configuration files DirectoryCfg.xml and DirectoryCfg.htm, you need to define the relevant security properties in IIS and Windows.

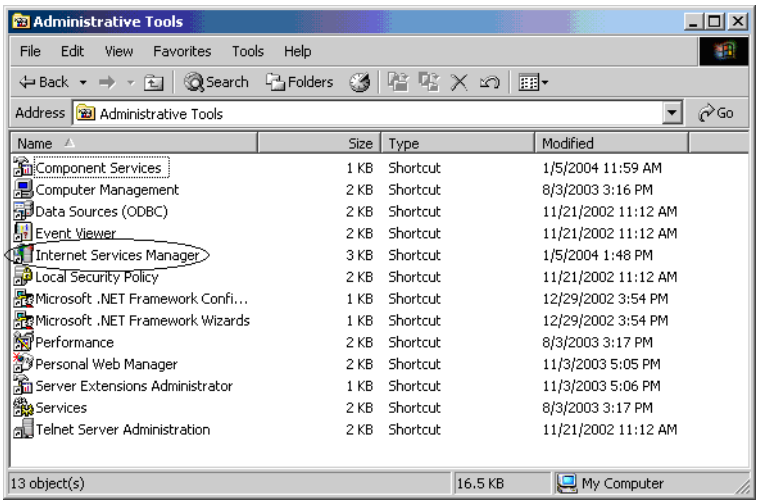
To define security properties in IIS:

1 Open the Windows *Control Panel* and double-click the **Administrative Tools** icon.



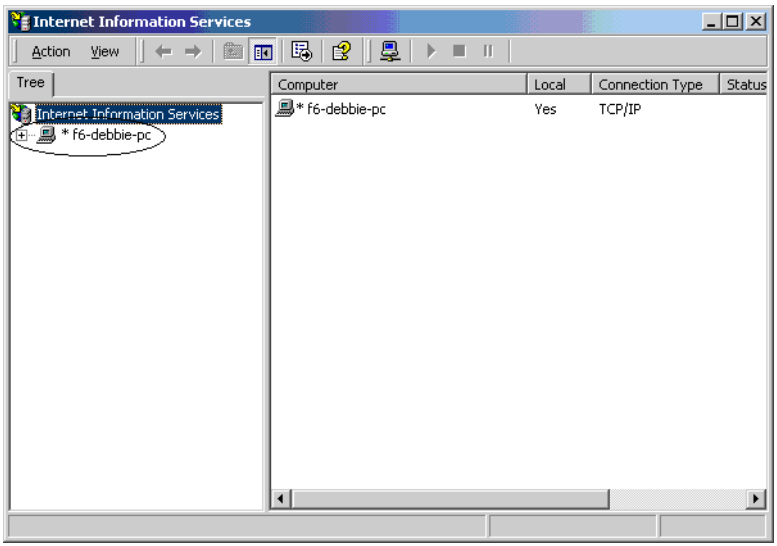
The *Administrative Tools* window opens.

2 Double-click the **Internet Services Manager** icon.



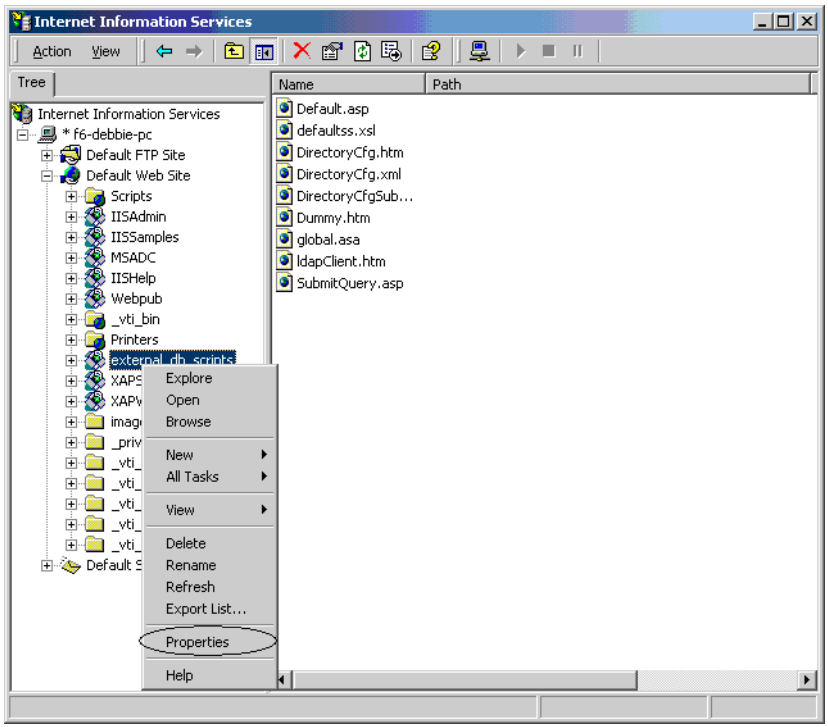
The *Internet Information Services* window opens.

3 Expand the server list.



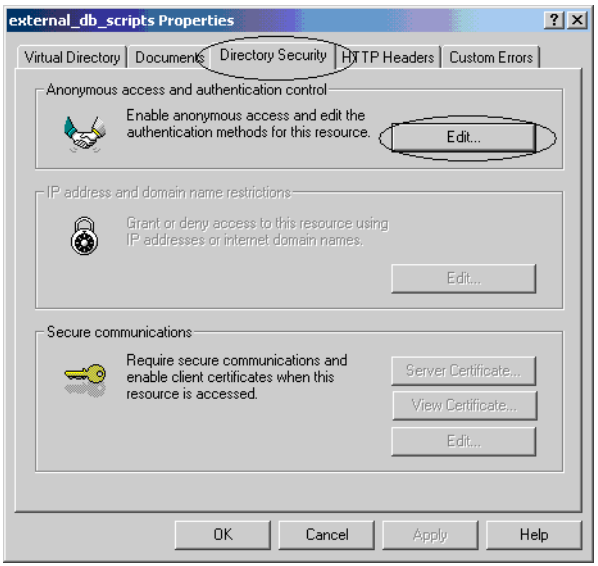
4 Expand the **Default Web Site** list.

5 Right-click the virtual web site you created for the script files in the previous procedure, and then click **Properties**.



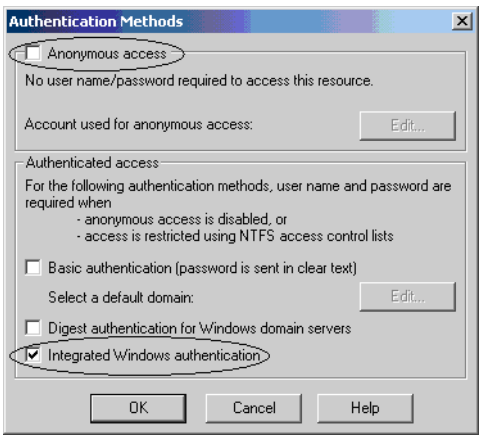
The *Virtual Directory Properties* dialog box opens.

6 Click the **Directory Security** tab, and then click the **Edit** button.



The *Authentication Methods* dialog box opens.

7 Perform the following settings:



— Clear the **Anonymous access** check box.
— Select the **Integrated Windows authentication** check box
— Click **OK.**

The system returns to the *Virtual Directory Properties* dialog box.

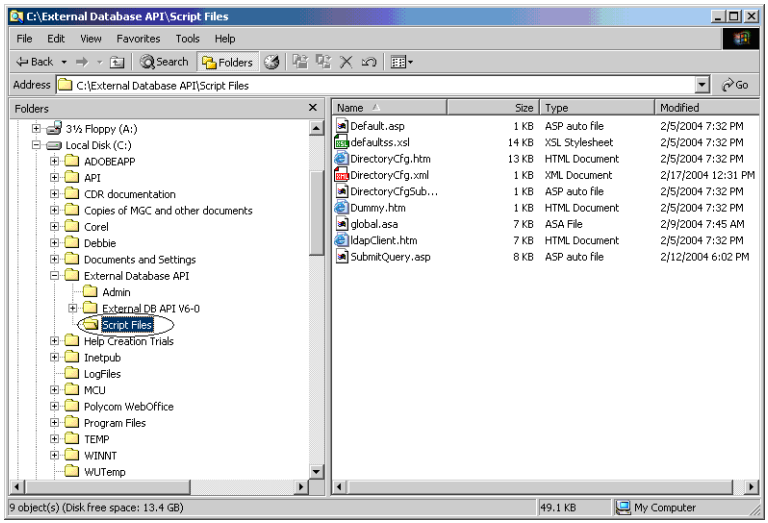
8 Click OK.

The system returns to the Internet Information Services window.

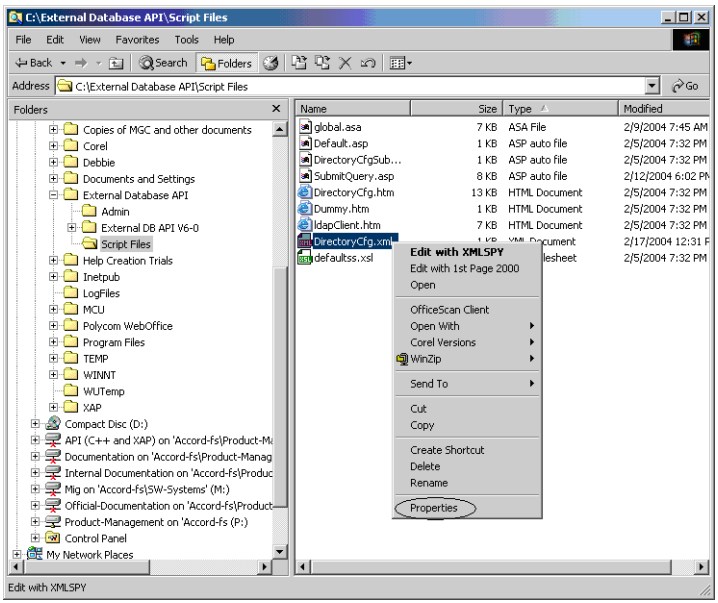
To define security properties in Windows:

1 Open Windows Explorer.

2 Select the directory to which you copied the sample scripts.



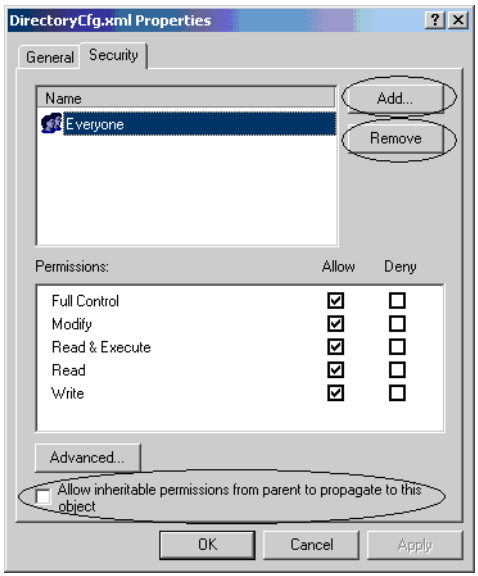
3 Right-click the **DirectoryCfg.xml** file, and then click **Properties.**



The *DirectoryCfg.xml Properties* dialog box opens.

4 Click the **Security** tab.

5 Remove existing permissions as follows:



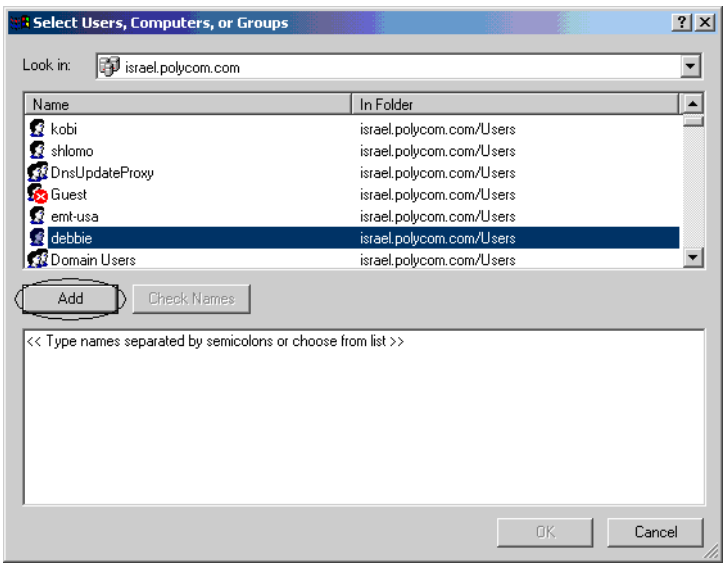
— If the **Allow inheritable permissions from parent to propagate to this object** check box is selected, clear it.

— If **Everyone** is listed in the *Name* pane, select **Everyone** and then click **Remove**.

6 Click **Add**.
The *Select Users, Computers, or Groups* dialog box opens.

7 In the upper pane, select the user who is to be authorized to access the DirectoryCfg.xml file.

8 Click **Add**.

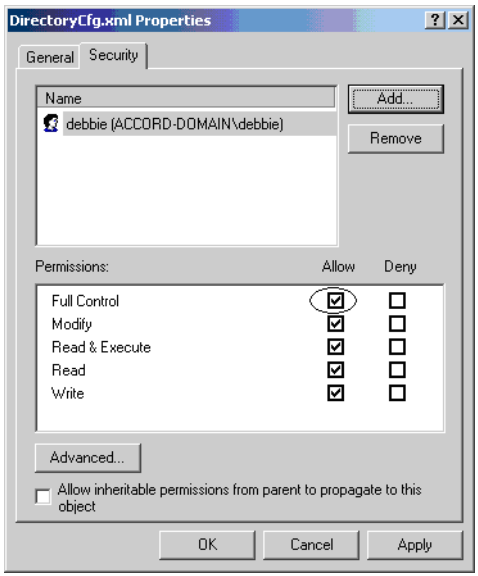


The selected user is displayed in the lower pane.

9 Click **OK**.

The system returns to the *DirectoryCfg.xml Properties* dialog box, with the selected user displayed in the *Name* pane at the top of the dialog box.

10 Select the **Full Control** check box in the *Allow* column, and then click **OK**.



The system returns to the Windows Explorer window.

11 Repeat steps 3 to 10 to define security properties for the **DirectoryCfg.htm** file.

Configuring the Directory Configuration File

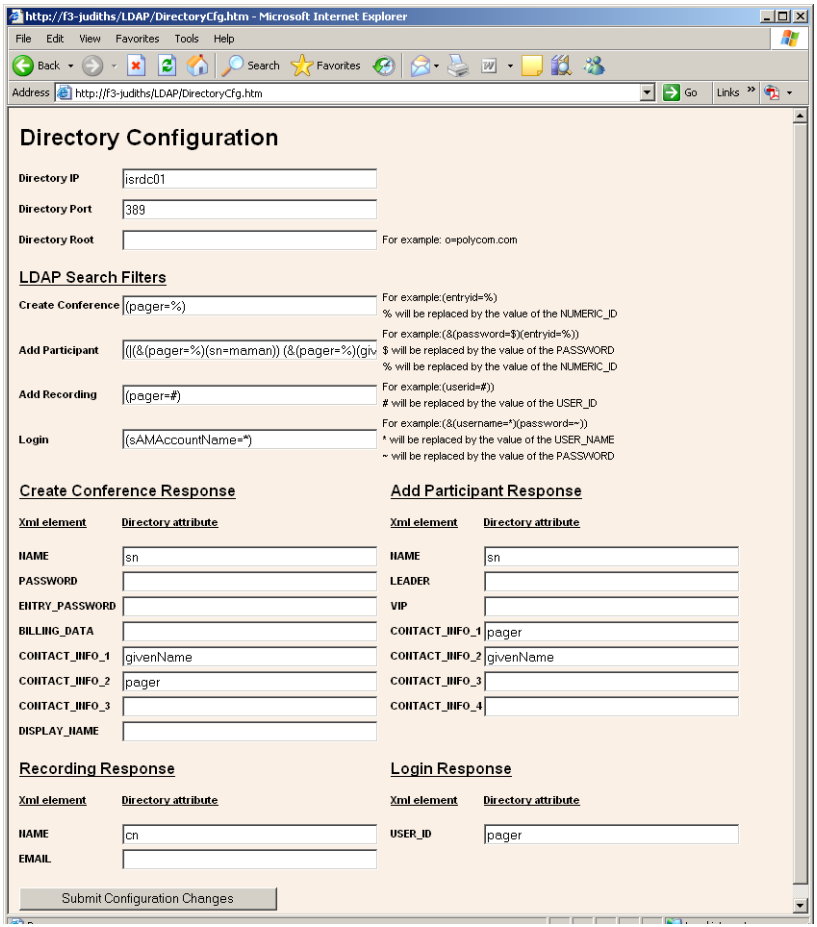
The directory configuration file, DirectoryCfg.XML, holds the following information:

- The directory IP address and other directory details.
- The LDAP search filters used to verify that a specific conference can be created, that a specific participant can join a specific conference and that a specific operator can log in to the MCU.
- A mapping of the directory attributes to the XML elements used in the MGC External Database API.

To update the directory configuration file:

1 Open the file **http://<URL>/<virtual directory name>/DirectoryCfg.htm**.

The *Directory Configuration* window is displayed.



2 Fill in the fields in the Directory Configuration window as follows:

Table 4-1 *Directory Configuration Parameters*

Field	Description
Directory Location Information	
Directory IP	The IP address of the computer where the directory is located.
Directory Port	The Port at which the directory is located.
Directory Root	The directory root.
LDAP Search Filters The search filters must comply with the LDAP dialect syntax. Refer to the relevant MSDN documentation for details of the LDAP dialect syntax if necessary. **Note:** Sometimes searching the directory takes a long time. To reduce the search time, add an index to the field.	
Create Conference	The search filter to be used to verify that a specific participant can create a conference with a specific ID. The participant is identified by E.164 (IP participants) or CLI number (ISDN participants). The simplest type of search filter will verify that the specified Conference ID exists in the directory. For example, if the name of the directory attribute to be checked for the Conference ID is **user_id**, then the search filter will be: (**user_id=%**). You can specify more complicated search filters as required.

Table 4-1 *Directory Configuration Parameters (Continued)*

Field	Description
Add Participant	The search filter to be used to verify that a specific participant can join a conference with a specific Conference ID with a specific password. The participant is identified by E.164 (IP participants) or CLI number (ISDN participants). The simplest type of search filter will verify that the specified password is valid for the conference with the specified Conference ID. For example, if the name of the attribute containing IDs is **user_id**, and the name of the attribute containing passwords is password, then the search filter will be: (**&(password=$)(user_id =%)**). You can specify more complicated search filters as required.
Add Recording	Only applicable to ReadiRecorder systems.
Login	The search filter to be used to verify that a specific user can log in to the MCU with a specific password. The simplest type of search filter will verify that the specified user is authorized to log in to the MCU with the specified password. For example, if the name of the attribute containing user names is username, and the name of the attribute containing passwords is password, then the search filter will be: (**&(username=*)(password=~)**). You can specify more complicated search filters as required.
Create Conference Response These fields identify the mapping between the directory attributes and the XML elements for a Create Conference response. The contents of the specified attributes are inserted in the XML response sent to the MCU. Most of the fields are optional. However, if conference or chairperson passwords are to be used in Ad Hoc conferences, it is recommended that they will be provided by the directory, otherwise they will be assigned randomly by the MCU.	
NAME	The attribute containing the name of the conference.
PASSWORD	The attribute containing the chairperson password for the conference.
ENTRY_ PASSWORD	The attribute containing the conference entry password.

Table 4-1 *Directory Configuration Parameters (Continued)*

Field	Description
BILLING_DATA	The attribute containing the billing code for the conference.
CONTACT_INFO_1	The attribute containing the text to be added in the first Info field (Info1) for the conference.
CONTACT_INFO_2	The attribute containing the text to be added in the second Info field (Info2) for the conference.
CONTACT_INFO_3	The attribute containing the text to be added in the third Info field (Info3) for the conference.
Add Participant Response These fields identify the mapping between the directory attributes and the XML elements for an Add Participant response. The contents of the specified attributes are inserted in the XML response sent to the MCU. Most of the fields are optional.	
NAME	The attribute containing the display name to be used for the participant.
LEADER	The attribute which indicates whether or not the participant is to be assigned as a conference chairperson. This attribute must contain the value true or false.
VIP	The attribute which indicates whether or not the participant is a VIP. This attribute must contain the value true or false.
CONTACT_INFO_1	The attribute containing the text to be added in the first Info field (Info1) for the participant.
CONTACT_INFO_2	The attribute containing the text to be added in the second Info field (Info2) for the participant.
CONTACT_INFO_3	The attribute containing the text to be added in the third Info field (Info3) for the participant.
CONTACT_INFO_4	The attribute containing the text to be added in the fourth Info field (Info4) for the participant.
DISPLAY_NAME	The attribute containing the display name of the conference.

Table 4-1 *Directory Configuration Parameters (Continued)*

Field	Description
Login Response	
The USER_ID field is not currently used. **Note:** The script does not return the authorization level from the database. If the operator is found in the database, then the script will return the value administrator in the AUTHORIZATION_GROUP element in the XML response sent to the MCU.	
Recording Response	
Only applicable to ReadiRecorder systems.	

3 Click the **Submit Configuration Changes** button.

4 Close and restart IIS in order to complete the configuration process.

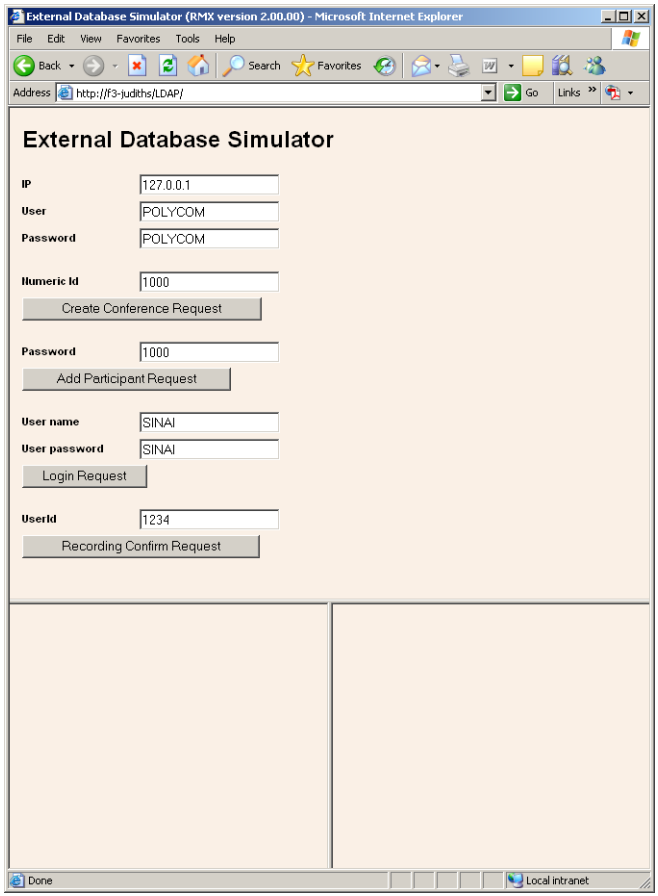
Testing the Directory Configuration File using the External Database Simulator

The virtual directory site contains a simulator that you can use to check that the attribute mapping you set up in the configuration file produces the correct results.

To use the simulator:

1 Open the file **http://<URL>/<virtual directory name>/default.asp**.

The *External Database Simulator* window opens.



2 Enter the IP address of the MCU in the *IP* field, and the MCU user name and password in the *User* and *Password* fields.

3 To test the *Create Conference* request:

— In the *Numeric ID* field, enter a Conference ID.

4 To test the *Add Participant* request:

— In the *Numeric ID* field, enter a Conference ID.

— In the Password field, enter a password.

5 To test the *Login* request:

— In the User name field, enter a user domain name

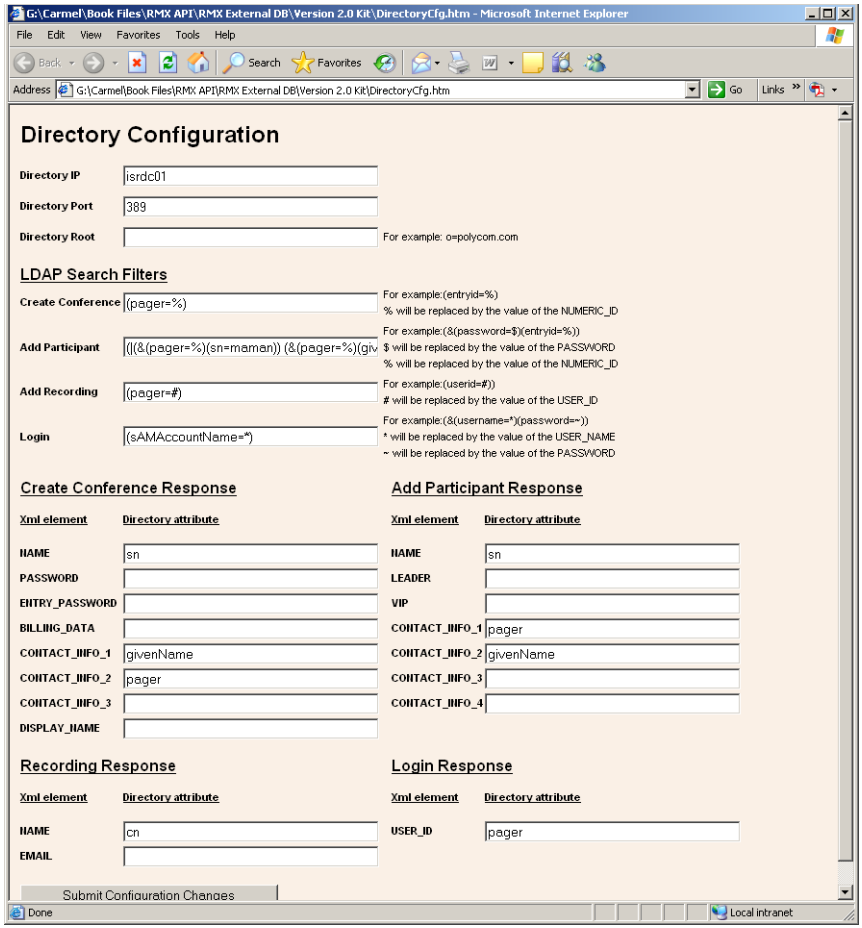
— In the User password field, enter a password.

6 Click either the **Create Conference Request**, the **Add Participant Request**, or the **Login Request** button.

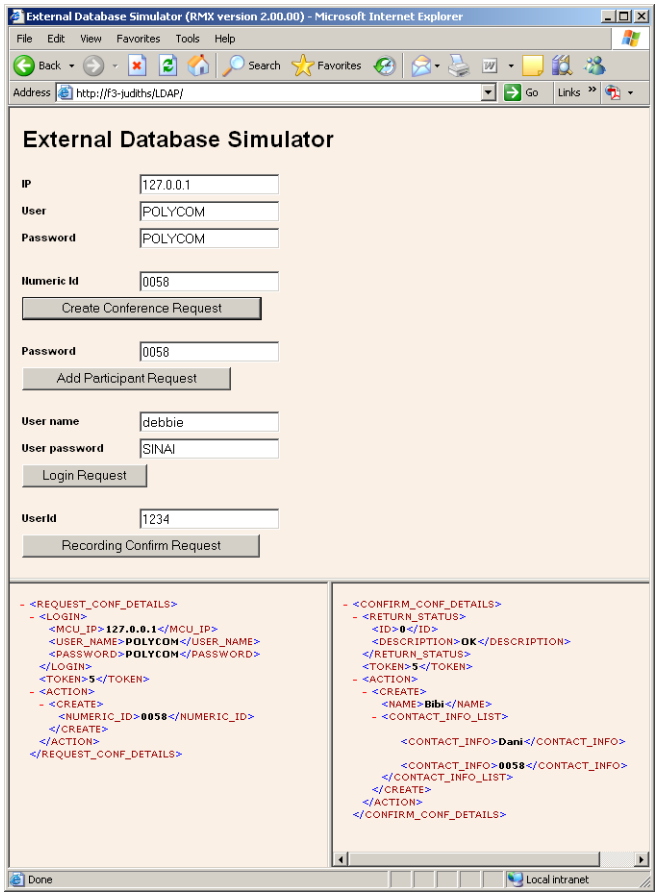
The bottom left pane displays the XML request sent from the MCU, and the bottom right pane displays the XML response sent back to the MCU.

Example

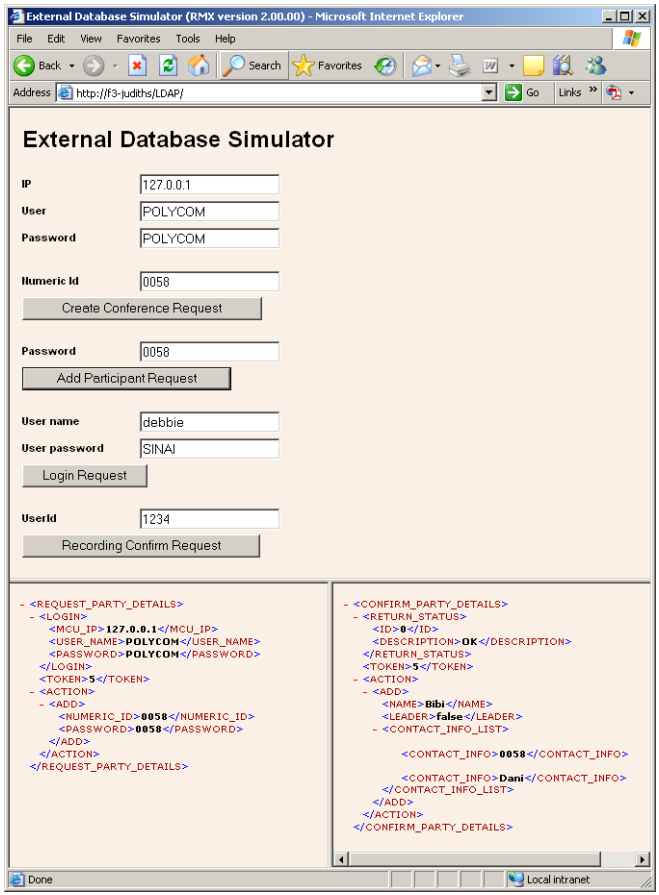
If the *Directory Configuration* file is set up as follows:



Clicking the **Create Conference Request** button will produce the following result:



Clicking the **Add Participant Request** button will produce the following result:



Clicking the **Login Request** button will produce the following result:

