




# **Bria 3.0 Provisioning Guide**

## ***Retail Deployments***

CounterPath Corporation.  
Suite 300, One Bentall Centre  
505 Burrard Street Box 95  
Vancouver BC V7X 1M3  
Tel: 1.604.320.3344  
sales@counterpath.com www.counterpath.com

© January 2010

This document contains information proprietary to CounterPath Corporation, and shall not be used for engineering, design, procurement, or manufacture, in whole or in part, without the consent of CounterPath Corporation.

Counterpath and the  logo are trademarks of Counterpath Corporation

CounterPath makes no warranty regarding the content of this document, including—but not limited to—implied warranties of fitness for any particular purpose.

In no case will CounterPath or persons involved in the production of this documented material be liable for any incidental, indirect or otherwise consequential damage or loss that may result after the use of this publication.

This manual corresponds to version 3.0 of Bria.

Microsoft Windows is a registered trademark of the Microsoft group of companies. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. Linux is a registered trademark of Linus Torvalds in the U.S. and other countries. Debian is a registered trademark of Software in the Public Interest, Inc. Apache is a trademark of The Apache Software Foundation. Pentium 4 is a registered trademark of Intel, Corp.

# Contents

1 About Provisioning .....	3
1.1 Provisioning Functions .....	3
1.2 What Provisioning Does: Writing to Settings.....	3
1.3 The Mechanism of Remote Provisioning .....	5
2 Bria Login Procedures .....	9
2.1 Choosing a Login Profile .....	9
2.2 About Account Credentials and Logging In .....	10
2.3 No Login .....	12
2.4 Local Profile .....	13
2.5 Remote Login Profiles .....	15
3 Updates and Upgrades .....	21
3.1 General Setup.....	21
3.2 Remote Update .....	23
3.3 Remote Upgrade .....	24
A Script Samples .....	27
B Macros .....	28



# About this Manual

This manual describes the *mechanism* of remote login/provisioning. It describes how to set up a server (or servers) for the remote login and optionally the remote provisioning, remote update and remote upgrade features of Bria:

- Remote login controls access to the application; the softphone will not start until the user has logged in.
- Remote provisioning lets you configure the softphone remotely.
- Remote update lets you change the configuration of a given deployment of Bria at runtime (outside of the login process).
- Remote upgrade lets you deploy upgrades of the software remotely.

This manual is intended for:

- Service providers who have purchased a branded and/or customized version of Bria.

This manual is intended to be read in conjunction with:

- “Bria 3.0 Configuration Guide - *OEM Deployments*” which describes the features that can be configured through remote provisioning.



# 1 About Provisioning

## 1.1 Provisioning Functions

Provisioning of Bria includes the following features:

- Controlling access to the VoIP service through a remote login. See page 9.
- The ability to provide a license key remotely. See page 9.
- Updating the Bria configuration (changing the factory defaults). Bria can be configured differently for each user. This feature is optional. See page 21.
- Providing upgrades to the executable by making new versions of Bria available to each Bria installation to download. This feature is optional. See page 21.

## 1.2 What Provisioning Does: Writing to Settings

Each provisioning function involves writing to settings stored on Bria computer. These settings control the behavior of various features of Bria. For example, a successful login request will result in the creation of new settings representing the account. A remote update may result in changing the value of existing settings.

For detailed information on settings and the features they control, see “*Bria 3.0 Configuration Guide - OEM Deployments*”.

### 1.2.1 Provisioned Settings Overwrite GUI Settings

Settings are assigned values in several ways:

- A setting has a default “factory” value.
- Some settings can be changed by the user on the GUI.
- Remote provisioning lets you can change the value of any setting.

At startup, the factory values are loaded, then the user overrides are loaded (overwriting factory values), and finally values that you send through the provisioning response are loaded (overwriting factory or user values). At shutdown, the current user overrides and provisioning overrides are persisted to the user file.

Keep in mind that provisioned settings override user settings. A user may complain that they change a value on the GUI but each time they restart Bria, their changes are lost: you are probably overwriting their value when you provision.

The Bria Settings reference documentation (a Microsoft® Excel® document) includes a column that identifies settings that are represented on the softphone GUI.

## 1.2.2 Syntax of Settings

Each setting has a fully qualified name: <domain>:<section>:<setting>

For example, proxies:proxy0:register.

The syntax for setting values via provisioning is:

<domain>:<section>:<setting>=<“value”>

For example, proxies:proxy0:domain=“domainA.com”

- The value of the variable must appear in double quotes.
- Always a string. True is represented by “true” or “1”. False is represented by “false” or “0”.
- The Bria process that interprets the settings ignores the case of the value (uppercase or lowercase), except for literals such as display names.

## 1.3 The Mechanism of Remote Provisioning

Each remote provisioning service involves an exchange between the login server and an individual Bria client. The exchange is performed over HTTP or HTTPS.

### 1.3.1 Servers

You must deploy servers to handle the provisioning requests:

- The “login server”: a server to handle login requests, if you decide to implement login. This server is simply a web server that, at a minimum, can serve one plaintext or XML file.
- The “update server”: a server to handle remote update.
- The “upgrade executable server”: a server to handle remote upgrades of the Bria application.

These server roles may in fact all be deployed on the same physical server: that is your decision.

The URLs for these servers are specified as follows:

- The login server is either set in your brand, discovered through DHCP, or manually entered by the user on the Login dialog. See “Remote Login Profiles” on page 15.
- The update server and upgrade executable server (if they are being used) are either set in your brand, or you set them by including them in the provisioning response that you set when the user logs in.

### 1.3.2 Bria-to-Server Exchange

The exchange between Bria and the appropriate server involves the following:

- When the appropriate trigger occurs, Bria sends an HTTP or HTTPS request to the server. For login, the trigger is the user pressing OK on the Bria login dialog. For remote upgrade, the trigger is startup of the softphone.
- The server responds.
- Bria reads the response and takes the appropriate action: starts the softphone and registers with the SIP proxy, or finds and installs the upgrade.

#### Use of Scripts and Macros

You may want to run an appropriate script on the given web server, to provide the information required by Bria. To run a script, include it in the URL for that server.

Running scripts usually requires information about the user’s deployment. The URL for the appropriate server can include macros. When Bria contacts the server, it replaces the macros with the real data and includes this information in the HTTPS request.

Your script must understand the names assigned to the macros.

For example a URL of

```
https://mycustomloginserver.com/login.php?platform=$platform$&lic=$license$
```

might become this POST used to log in the user:

```
https://mycustomloginserver.com/login.php
```

```
-----  
Username=21187  
Password=rosebud  
platform=win32  
lic=d3874ihfd8t23975v1iu5182ruity3iusapor236u545uye0r9qwjj
```

Note that “Username” and “Password” (with initial capitals) are always sent in a login POST; the URL does not have to include macros for this data.

See “Script Samples” on page 27 for samples of some of the scripts that are mentioned in this manual.

See “Macros” on page 28 for a list of macros that Bria supports.

### 1.3.3 Communication Mechanism

All communications between Bria and the login server are performed over HTTP or HTTPS, as follows:

- Custom login uses POST.
- Remote update and remote upgrade use GET.

The remote provisioning mechanism does not support redirect.

If using HTTPS, you need a trusted certificate (not self signed). Bria will only accept certificates whose authenticity can be verified through the trust chain.

## 1.3.4 Data Format

All the data included in the GET or POST response is in a specific format. This format is similar to that of Microsoft® Windows® .ini files.

The information is organized into three portions, which must appear in this order:

- [DATA]
- [SETTINGS]
- [##MEMORY##]

### Example

```
[DATA]
Success=1
[SETTINGS]
proxies:proxy0:display_name="kokila"
proxies:proxy0:enabled="1"
proxies:proxy0:username="kperera"
proxies:proxy0:password="dfher43d89dhferuieo98375uy8"
proxies:proxy0:domain="domainA.com"
```

#### [DATA]

This section contains the response to requests:

Success=<value>, a boolean. This data is required.

Failure=<message>, which is optional if the success is 0. For login, the string you enter here will be displayed in the Login dialog.

#### [SETTINGS]

This section contains settings to be written to persistent memory. The values will be used immediately.

At shutdown, these settings will be written to the local settings file on the Bria computer.

#### [##MEMORY##]

This section contains settings to be written to non-persistent memory. The values will be used immediately, but only for the current session.

At shutdown, these settings will not be written to the local settings file.

#### CRLF

The response must end with a CRLF. If this is missing, the last line of the response is ignored.

### Handling and Encryption of Passwords

All “password” settings in any domain/section are handled as follows:

- Bria does not interpret passwords in any way, so the value the login server passes to Bria can be encrypted.
- Bria encrypts the value before storing it, regardless of whether or not it is already encrypted. When a stored value is read in order to pass it to the login server, it is first decrypted.
- When a password that the user has been entered into a dialog is then passed to the login server, Bria does not encrypt the value.

## 1.3.5 Example of an Implementation

The hardware requirements of the login server depend on what the server will do. If it will have a complicated backend database and processing in order to retrieve the settings that are to be provisioned, then the server should be of higher processing capabilities. Regardless, the login server is simply a web server and it only needs to serve one file for provisioning; this file is in plaintext or XML format.

The login server could be a Linux® machine with an Apache™ web server or a Microsoft® Windows® machine with an IIS web server.

For their internal deployment, CounterPath uses Debian® Linux with Apache version 2. The login server is a Pentium® 4 with 3GHz processor. This server scales to thousands of requests per second. We use the internal database of the SIP proxy (this can be a MySQL® database) which contains all usernames and passwords. The provisioning response is constructed based on login information retrieved from Bria via the login PHP script.

# 2 Bria Login Procedures

## 2.1 Choosing a Login Profile

This manual describes how to implement provisioning using one of the following profiles.

Profile	User login	Description	See
No login	No	The user obtains account credentials (credentials for either a SIP or an XMPP account) outside of Bria. The user never logs in.	page 12
Local profile	Yes, in “local mode”	Local login. The user obtains account credentials outside of Bria. The user logs into Bria using the account credentials. Bria automatically uses these credentials to set up the account.  It is not possible to provision settings (other than the account credentials) differently for each user.	page 13
Remote login profiles: <ul style="list-style-type: none"><li>• DHCP profile</li><li>• Manual Configuration profile</li><li>• Hard-coded profile</li></ul>	Yes	Remote login. The user obtains account credentials outside of Bria. The user logs into Bria using these login credentials. The credentials are sent to the login server, which sends back account credentials and other settings. Bria automatically uses these account credentials and settings to set up the account.  These profiles support provisioning of the license key.  It is possible to provision different settings for each user.	page 15

## 2.2 About Account Credentials and Logging In

It is important to understand the difference between account credentials and login credentials.

### SIP Account Credentials

The SIP account credentials allow the user to register for your VoIP service; they are known to your SIP registrar. These credentials are user name, password, and the optional authorization user name.

These credentials are represented in Bria by settings in the proxies domain. For more information on these settings, see the Bria Settings list.

### XMPP Account Credentials

The XMPP credentials allow the user to access the XMPP server; they are known to the XMPP server.

These credentials are user name and password. These credentials are represented in Bria by settings in the proxies domain. For more information on these settings, see the Bria Settings list.

### Login

Login refers to the process of signing into the VoIP service. The Bria user must enter login credentials in order to access to Bria. The login credentials are user name and password. Typically, only the user name (or authorization name, if used) and the password are used to authorize the login request.

### Purpose of Credentials

Credential	Comment
User name	<ul style="list-style-type: none"> <li>The login user name is meaningful to the user (for example, their own name).</li> <li>The account user name follows the syntax for your accounts – it may be a number or words.</li> </ul>
Authorization user name.	<p>The authorization user name is optional in all profiles.</p> <p>An authorization user name is useful, for example, if you allow usernames that are short and therefore easy to guess. The authorization user name is used in place of the user name to register the account with the proxy. It provides an added layer of security.</p> <p>The authorization user name is used only to register the account. It does not replace the account user name in identifying the user to the outside world.</p> <p>If you use an authorization name, make sure it is different from the user name!</p>
Password	

## Comparison of Login and Account Credentials

This table compares the account credentials and login credentials.

	<b>Account credentials</b>	<b>Login credentials</b>
Are the credentials required?	<ul style="list-style-type: none"> <li>Always required</li> </ul>	<ul style="list-style-type: none"> <li>Used only if you decide to use the Bria login process.</li> </ul>
Purpose of credentials	<ul style="list-style-type: none"> <li>Let the user register/access the account</li> </ul>	<ul style="list-style-type: none"> <li>The username and password are used to validate the user's request to log into your VoIP service.</li> </ul>
How credentials are created	<ul style="list-style-type: none"> <li>Account credentials are generated on your side</li> </ul>	<ul style="list-style-type: none"> <li>Login credentials are generated on your side and sent to the user outside of Bria. Typically, all the login credentials are meaningful to the user. The user enters them in the Login dialog and Bria sends them to the login server.</li> </ul>
How Bria gets the credentials	<ul style="list-style-type: none"> <li>For Local profile, entered by user in the login dialog or the Account Settings window.</li> <li>For the three remote login profiles, sent to Bria during a login request.</li> </ul>	<ul style="list-style-type: none"> <li>Entered by the user in the login dialog.</li> </ul>
How the credentials are saved on the Bria computer	<ul style="list-style-type: none"> <li>When Bria receives account credentials, these credentials are written to the settings file on the Bria computer.</li> </ul>	<ul style="list-style-type: none"> <li>Login credentials are written to the settings file only if the "Remember" boxes on the dialog are checked.</li> </ul>
How the credentials are changed	<ul style="list-style-type: none"> <li>Can be changed through provisioning</li> </ul>	<ul style="list-style-type: none"> <li>Login credentials cannot be changed through provisioning.</li> </ul>

## 2.3 No Login

With this profile:

- You do not need a login server.
- The user can have more than one account.

Notify CounterPath that you do not want any login. Your version of Bria will be configured for this mode.

You must provide the user with account credentials. The password should not be encrypted, because the user will enter it manually.

The user must display the Account tab in the Account Settings window, enter the account credentials, and enable the appropriate account or accounts. These credentials are written to the account credential settings on the Bria computer. Bria then registers with the SIP registrar using these credentials.

Each time Bria is started, Bria will immediately register every enabled account with the SIP registrar.

## 2.4 Local Profile

This profile is appropriate only if you, the service provider, do not need to change any Bria settings on a per-install basis. In other words, the factory settings (other than account credentials) are suitable for all users. Local login provides control in situations in which a computer is shared: Access to Bria is controlled by login, and when the user logs in, their individual data (account credentials, contacts, and so on) are loaded

The user obtains account credentials outside of Bria. The user logs into Bria using the account credentials. Bria automatically uses these credentials to set up the account.

With this profile:

- You do not need a login server.
- The user is restricted to one account.
- You need a mechanism outside of Bria (for example, a webpage) for creating user account credentials.

### 2.4.1 Credentials Required

You must provide the user with account credentials.

- The account user name and login user name must be identical.
- The account password and the login password must be identical. The password should not be encrypted, because the user will enter it manually.
- The authorization user name is optional. If you decide to use it, you must ask CounterPath to include it in the Login dialog (by default, this dialog does not show this field).
  - An authorization user name is useful, for example, if you allow usernames that are short and therefore easy to guess. The authorization user name is used in place of the user name to register the account with the proxy. It provides an added layer of security.
  - The authorization user name is used only to register the account. It does not replace the user name in identifying the user to the outside world.
  - If you use an authorization name, make sure it is different from the user name!

### 2.4.2 How the User Will Log in

#### First-time Login

The first time the user starts Bria, the “local login” Login dialog is displayed. The user enters the account credentials in the Login dialog. (In other words, the account credentials are used as the login credentials). These credentials are written to the account credential settings on the Bria computer. Bria then registers with the SIP registrar using these credentials.



The Login dialog may also include a “Display name” dialog

## Subsequent Logins

The next time the user starts Bria, the Login dialog appears and proceeds in one of these ways:

- If the user had checked the “Remember login information” and “Log in automatically” boxes during the last login, then login starts immediately.
- If the user had not checked these boxes, login starts only when the user clicks OK.

## 2.4.3 Setting up for Local Mode Login

Notify CounterPath that you want to implement the local mode login. Your version of Bria will be configured for this mode.

You can also speak to your account representative about customizing the login dialog.

## 2.5 Remote Login Profiles

There are three types of remote login profiles, that is, profiles that support remote profile:

- DHCP profile. Bria detects the login server URL using DHCP discovery.
- Manual Configuration profile. The user enters the login server URL on the Login dialog.
- Hardcode URL profile. Your brand of Bria is hard-coded with the login server URL. With this profile, if you change the URL, you must request a new brand of Bria.

Once Bria has obtained the login server URL, the process is the same for all these profiles.

The user obtains login credentials outside of Bria. The user logs into Bria using these login credentials. The credentials are sent to the login server, which sends back account credentials and other settings. Bria automatically uses these account credentials and settings to set up the account.

It is possible to provision different settings for each user.

With this profile:

- You need a login server.
- You need a mechanism outside of Bria for creating account credentials for the user. This mechanism may be a webpage.
- You can allow the user to have more than one account.

### 2.5.1 Credentials Required

You must provide the user with login credentials. You do not give the user the account credentials; instead, these credentials will be sent down through provisioning.

- The account user name and login user name can be identical or different.
  - The login user name is meaningful to the user (for example, their own name).
  - The account user name follows the syntax for your accounts – it may be a number or words.
- The account password and the login password are typically different for security reasons.
  - The login password should not be encrypted, because the user will enter it manually.
  - The account password does not have to be human-readable.
- The authorization user name (one of the account credentials but not a login credential) is optional.
  - An authorization user name is useful, for example, if you allow usernames that are short and therefore easy to guess. The authorization user name is used in place of the user name to register the account with the proxy. It provides an added layer of security.
  - The authorization user name is used only to register the account. It does not replace the account user name in identifying the user to the outside world.
  - If you use an authorization name, make sure it is different from the user name!

## 2.5.2 How the User Will Log in

### First-time Login

You provide login credentials outside of Bria. The first time the user starts Bria, the Login dialog is displayed. The user enters the login credentials in the Login dialog and presses Login. The exchange proceeds as described in page 17.

### Subsequent Logins

The next time the user starts Bria, the Login dialog appears and proceeds in one of these ways:

- If the user had checked the “Remember login information” and “Log in automatically” boxes during the last login, then login starts immediately.
- If the user had not checked these boxes, login starts only when the user clicks Login.

### Reinstalls

If the user ever needs to start Bria afresh (for example, on a different computer that has no Bria settings), the exact same process as for the first-time start is followed.

## 2.5.3 Skipping Login

With the DHCP profile, Bria has fallback behavior if the DHCP server cannot be reached, for example, when the user is not in the office. If the user has already successfully logged in at least once, then Bria skips login and starts Bria using the account credentials and other settings that are currently stored on the Bria computer.

Note that skip login only applies if the DHCP server cannot be reached, meaning that the login server URL cannot be obtained. Skip login does not apply if the login fails, for example because the login credentials were entered incorrectly or because the login server is not online.

Note that skip login is not necessary for the Manual Configuration or Hard-coded URL profiles because the login server URL can always be obtained.

## 2.5.4 Branding Bria for the Desired Profile

### DHCP Profile

- Advise your account representative that you want to support this profile.
- You can also speak to your account representative about customizing the login dialog.
- Specify if you want to limit the number of times the user can skip login (see above). For example, if you limit the skips to 3, then once the user has skipped login three consecutive times, the user will not be able to start Bria again until they have successfully logged in.

### Manual Configuration Profile

Advise your account representative that you want to support this profile.

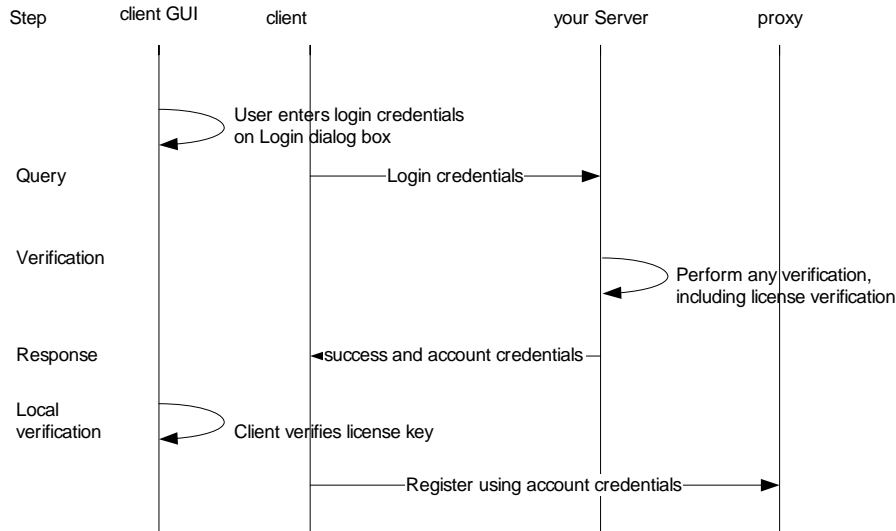
### Hard-coded URL Profile

- Advise your account representative that you want to support this profile.
- Provide the URL of the login server.

You can also speak to your account representative about customizing the login dialog.

## 2.5.5 How Login Proceeds

The login procedure is identical for all three remote login profiles. The login server must be set up to handle the following procedure.



### Login Procedure Is Invoked

The Log dialog is displayed. (You can speak to your account representative about customizing the login dialog.) The user enters the required information and presses Login.



## Query Step

Bria sends the data from the Login dialog. The data is encoded application/x-www-form-urlencoded.

The data is sent to the login server (the server specified in feature:custom\_login:server) in an HTTP POST. The value will be blank if the branded Login dialog does not include the corresponding field; this is not an error.

For example a URL of

```
https://mycustomloginserver.com/login.php?platform=$platform$&lic=$license$
```

might become this POST used to log in the user:

```
https://mycustomloginserver.com/login.php
-----
Username=21187
Password=rosebud
platform=win32
lic=d3874ihfd8t23975v1iu5182ruity3iusapor236u545uye0r9qwjj
```

where:

- “Username” and “Password” (with initial capitals) are always sent in a login POST; the URL does not have to include macros for this data.
- platform and lic are macros used by the login script; see “Use of Scripts and Macros” on page 5.

## License Key Management

You should set up Bria to include the license key in the data sent to the login server. There are two ways to send this data:

- Include one of the license macros in the URL. See page 28.
- Set the setting feature:custom\_login:always\_include\_license\_in\_post to true.

## Verification Step

The login server should perform any suitable verification on the sent data, according to your business rules.

Typically, this verification will include one of the following checks on the license key:

- For a new deployment (no license key was included in the query), determine that the query is valid, and if so, obtain a license key to send back to the user.
- For an existing deployment (the license key was included in the query), do nothing.

## Response Step: Failure

If there is a problem with any of the data, your server should return failure data in the following format:

```
[DATA]
Success=0
Failure="<message> "
<CRLF>
```

## Response Step: Pass

If your server can handle the request, it should return a success message and the account credentials. It can also return other settings that can be specified only at login.

Example with a license key passed in the SETTINGS section:

```
[DATA]
Success=1
[SETTINGS]
proxies:proxy:user_name="KPerera"
system:license:key="e48jey45379ryeioo8a7e934q8dhfudufoladskiuwb"
[##MEMORY##]
proxies:proxy0:password="rosebud"
<CRLF>
```

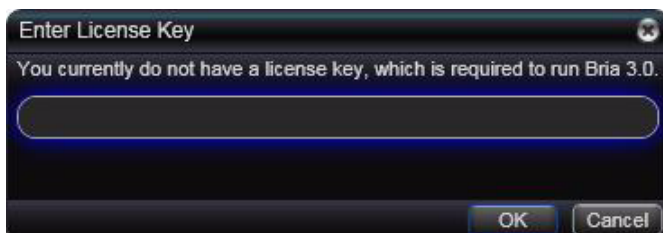
where:

- success: this line is required.
- Settings: the username will be saved at shutdown.
- ##Memory##: the password will not be saved at shutdown.
- The response must end with a CRLF.

## Local Verification

Bria next takes one of these actions, depending on the response received from the server:

- If the response was a failure, then the Login dialog appears again. The process goes back to “Login Procedure Is Invoked” on page 17.
- If the response was a success, then Bria verifies that the license key is valid. The license key is whatever is currently stored locally: it could be the license key that was sent down in the response, or it could be the license key that was already stored locally at startup, or it could be empty.
  - If the key is valid, Bria starts.
  - If the key is not valid or is empty, then the Enter License Key dialog appears. When the user enters the license key (obtained outside of Bria, for example in an email sent to all new customers), the Bria verifies that the entered license key is valid. If the key is valid, Bria starts.





# 3 Updates and Upgrades

## Remote Updates

You can configure Bria to check with the update server at specified intervals for changes to the user's settings.

## Remote Upgrades

You can make software upgrades of Bria available on a web server. Bria can be set up to check with this upgrade executable server for software upgrades. If an upgrade is available, the user is prompted to download and install it.

## 3.1 General Setup

When you branded your softphone, you provided CounterPath with the values for the following settings. Keep in mind that if later find that the values you provided are not suitable, you can change the values in the same way as you change any setting: using remote provisioning.

Domain:Section	Setting	Comment
feature:auto_update	code_server_url	The web server for remote upgrades of the executable. Default is empty.
feature:auto_update	config_server_url	The web server for remote update. Default is empty.
feature:auto_update	block_timer_t3_s	See below for a description. Default is 10 seconds. Typically, leave the default.
feature:auto_update	deffer_timer_t2_s	See below for a description. Default is 60 seconds. Typically, leave the default.
feature:auto_update	update_check_initial_t1_s	See below for a description. Default is 20 seconds. Typically, leave the default.
feature:auto_update	update_check_t1_s	See below for a description. Default is 86400 seconds (24 hours). Typically, leave the default.
feature:auto_update	timer_factor	See below for a description. Default is 1.00

## Timer Settings

Remote upgrades and remote updates rely on four timers in the user's settings. The timers control how frequently Bria contacts the update and upgrade executable servers.

All values are in seconds. You can use the `timer_factor` setting to convert the values on your side into seconds.

- `update_check_initial_t1_s`.
- `update_check_t1_s`.
- `deffer_timer_t2_s`.
- `block_timer_t3_s`.

Automatic checks for remote upgrades and automatic checks for remote updates are performed at the same point: when the user starts Bria. The interaction of the timers occurs as follows:

1. Bria starts.
2. The timer `update_check_initial_t1_s` starts.
3. When `update_check_initial_t1_s` expires, Bria checks its state of business (whether or not the user is busy with a call or instant message session).
  - If Bria is busy, `deffer_timer_t2_s` starts. When this timer expires, Bria checks its business again. `deffer_timer_t2_s` continues restarting and expiring until Bria is no longer busy (when the user hangs up from an active call).
  - If Bria is not busy, it contacts the servers.
4. The timer `block_timer_t3_s` is set when a check is initiated at Bria. Another check is not allowed as long as `block_timer_t3_s` is still active. This timer ensures that provisioning checks are not performed too often, and is especially useful for protecting against potential hacker requests (which may arrive with frequency). The timer `block_timer_t3_s` is typically shorter in duration than `update_check_t1_s`.
5. After the first check, the cycle starts over at step 1 using the timer `update_check_t1_s` (not `update_check_initial_t1_s`).

## Changing Timer Settings

New timer settings will take effect as follows:

- If the timer is not running when the server sends new settings (and they are saved on the Bria computer), then the setting take effect immediately. The next time the timer is loaded, the new setting will be used.
- If the timer is running, the new setting takes effect after the timer expires and is reloaded. This means if `update_check_t1_s` still has 23 hours to go, it will be changed only after 23 hours. However, if the session is restarted, the new setting will take effect.

## 3.2 Remote Update

### 3.2.1 Setting Up

- Set up Bria as described on page 21.
- Set up the update server to handle the procedure described below.

### 3.2.2 How Remote Update Is Performed

Assuming that the timers are not all set to zero, this procedure runs “in the background” for as long as Bria is running.

1. When triggered by the timer, Bria checks for remote updates by sending a GET to the update server

For example, the value of `feature:auto_update:config_server_url` might be:

```
https://myupdatesettingsserver.com?language=$language&&build=$build&&name=$loginame$
```

This URL could result in a GET to your web server of:

```
myupdatesettingsserver.com?language=EN&build=16835&name=kperera
```

2. The update server must response with the following:

```
[DATA]
Success=0
<CRLF>
```

or

```
[DATA]
Success=1
[SETTINGS]
feature:auto_update:update_check_t1_s="3600"
<CRLF>
```

where:

- success: 1=true (there are updates) or 0=false (there are no updates).
- The [SETTINGS] section contains the changed settings. See “Data Format” on page 7 for details.
- The response must end with a CRLF.

## 3.3 Remote Upgrade

### 3.3.1 Setting Up

- Set up Bria as described on page 21.
- Set up an upgrade server as follows:
  - You can use a script to include logic that determines a given deployment needs an upgrade. See below for an example. Obtain the sample upgrade script from CounterPath and modify it to suit your needs. Or you can skip the script and manually set up your upgrade server to simply provide a success response when an upgrade is available and a failure response at other times.
  - If you are using scripts, set the URL for the upgrade server to include the script and any macros (for example, the language and the build macros).
  - When you want to deploy an upgrade, place it on the “upgrade location”.

### 3.3.2 How Remote Upgrade Is Performed

Assuming that the timers are not all set to zero, this procedure runs “in the background” for as long as Bria is running.

#### **Bria Sends a GET**

When triggered by the timer, Bria checks for available upgrades by sending a GET to the upgrade executable server.

- For example, if you are using scripts, the value of feature:auto\_update:code\_server\_url might be:

```
https://executablegradeserver.com/exe_upgrade.php?build=$build&language=$language&name=$loginame$
```

This URL could result in a GET to your webserver of:

```
https://executablegradeserver.com/exe_upgrade.php?build=38740&language=USEnglish&name=kperera
```

- Or if you are not using scripts, the value is simply the URL of the upgrade server:

```
https://executablegradeserver.com
```

#### **Server Response**

The upgrade executable server must respond with the following:

```
[DATA]  
Success=0  
<CRLF>
```

or

```
[DATA]
Success=1
Mandatory=1
version=60000
url=https://executableupgradeserver.com/newversion.exe
<CRLF>
```

where:

- `Success`: 1=true (there is an upgrade) or 0=false (there is no upgrade).
- `Mandatory`: 1=true. This response is optional; the default is “0”. Bria handles the upgrade differently depending on this response; see below.
- `version`: identifies a build stamp set by Bria during build time. Bria uses this version to determine whether to prompt the user to install the upgrade; see step .
- `url`: the absolute path to the installer software for the new version.
- The response must end with a CRLF.

The response **cannot** include a [SETTINGS] section. In other words, none of the user’s current settings can be changed via this response.

If no upgrades are found, Bria will recheck periodically for available upgrades. See “Timer Settings” on page 22 for details.

## Handling of the Upgrade

If an upgrade is available, Bria compares the build number of the application on the user’s computer to the build number specified in the response (60000 in the above example).

- If the response has the same number, Bria does not prompt the user to download
- If the response has a *different* number, Bria prompts the user to download the upgrade.
  - If the user initiates the download, Bria will download the installer and save it to the local Bria program folder. Bria will also prompt the user to exit in order to install the new version. The user can install immediately or postpone installation.
  - If the user declines the upgrade and the upgrade is optional, Bria will enter its timing cycle and display the download prompt again at the appropriate time. See “Timer Settings” on page 22.
  - If the user declines a mandatory upgrade, Bria shuts down
  - If the user declines with “do not ask me again” (possible only with an optional upgrade), Bria will not check again for upgrades during the session.

## “Install Later” Handling

If the user declines to install the downloaded version, then the next time Bria is started the user will be prompted to install the newer version. One of the following will occur:

- If the user initiates the installation, Bria will install the new (local) version.
- If the user declines, Bria will start the original version and will enter its timing cycle, displaying the download prompt again at the appropriate time. See “Timer Settings” on page 22.
- If the user declines with “do not ask me again”, Bria will start the original version and will not prompt to install again during the session.

Bria starts the version installed most recently. The automatic check scenario will be initiated as described in the previous section. The downloaded installer will not be deleted, to enable manual rollback, if required.



---

# A Script Samples

Contact CounterPath to obtain sample scripts.

These sample scripts, written in PHP, are intended to illustrate a possible implementation. They are not intended to be used without modification. You should write scripts suitable to your environment, in an appropriate scripting language.

## **login.php**

Custom login script. Bria passes in the username and password. After verification, if the login credential is correct, the server will write the proper settings into the response and send it back to Bria.

See “Use of Scripts and Macros” on page 5 for an example that uses this script.

## **exe\_upgrade.php**

Bria passes in the buildstamp. You may want to revise the script to also pass in the platform. It returns success is true or false, plus the URL where the upgrade of the Bria executable is located.

See “Remote Upgrade” on page 24.

# B Macros

Macro	Description	Value
\$acc_passwdn\$	where n is an account. The password for the specified SIP account (for deployments that support more than one SIP account). Stored as a setting.	
\$acc_usern\$	where n is an account. The username for the specified SIP account (for deployments that support more than one SIP account). Stored as a setting.	
\$build\$	The unique buildstamp.	For example, 12345
\$company\$	The company long name specified for your brand.	
\$computerid\$	Unique ID for this computer	
\$computername\$	From the operating system	
\$hashlicense\$	A hash of the license key. This allows the license key to be sent in a secure way	
\$IP\$	The IP address of this computer	
\$language\$	The language of the installed application.	USEnglish, French, German, Italian, Portuguese-Brazil, Spanish,
\$license\$	The license key.	
\$loginauthname\$	The login authorization name, if used. This is the name the user enters in the Login dialog. See page 10.	
\$loginname\$	The login username. This is the username the user enters in the Login dialog and is not necessarily the same as the SIP username. See page 10.	
\$loginpassword\$	The login password. This is the password the user enters in the Login dialog and is not necessarily the same as the SIP password. See page 10.	
\$MAC\$	The MAC address of the machine running Bria.	
\$platform\$	The operating system platform.	win32, mac, UNDEFINED.
\$product\$	The product name specified for your brand.	
\$release\$	The two-digit release.	For example, 2.5.
\$winusername\$	The Microsoft Windows user name	
\$winversion\$	The Microsoft Windows version.	WINNT4, WIN2K, WINXP, WINVISTA, OTHEROS.