

# Instant Roku Channel Device Registration and Linking

Version 2.02

12/03/2012

## Overview

1. Instant Roku Channel's implementation of Registration and Linking is a modified version of the process described in the Roku document "Device Registration and Linking".
2. Amazon S3 is used to communicate between the Roku player and the registration server. There is never a direct server-to-player connection. This has an advantage in that a Roku player will continue to operate even if there is a server outage, as the post-registration process is only dependent on a file being present in an S3 bucket.
3. The channel is pre-configured using the Instant Roku Player platform with the following Registration and Linking constants:

A Registration Partner ID string.

Number of characters in the Registration code from 3 to 6.

A polling retry interval from 1 to 999 seconds.

A polling retry duration from 1 to 9999 seconds.

The name of the Amazon S3 Registration bucket.

The name of the Amazon S3 Linking bucket.

The Access Key Id and the Secret Access Key for the buckets.

Various theme items such as colors and message text.

## Pre-Registration

1. The channel (the script running on the Roku player) generates a 3 to 6 character base-32 Registration Code and a 16 character base-32 Device Token:

Example Registration Code: AW2T

Example Device Token: 9UR4TE54FWFBMFHB

2. The channel writes a Registration Request file to the Amazon S3 Registration bucket using a name built from the Registration Partner ID string and the Registration Code. The file contains a variety of XML elements, most importantly the Device Token.

### Example Registration Request File

File Name: mycorp/AW2T

```
<?xml version="1.0" ?>
<registrationRequest>
  <deviceID>K0A09T000000</deviceID>
  <deviceTypeID>2100X</deviceTypeID>
  <firmwareVersion>013.01E01017A</firmwareVersion>
  <partnerID>mycorp</partnerID>
  <channelID>54d722a7-1f77-4e5b-8123-e9c1105c8c7c</channelID>
  <deviceToken>9UR4TE54FWFBMFHB</deviceToken>
  <requestTime>2012-06-23T22:03:13.411Z</requestTime>
</registrationRequest>
```

3. If the channel is unable to write the Registration Request file, for example because of a network problem or if the bucket keys are incorrect, an implementation dependent error message is displayed to the user.
4. If the Registration Request file is successfully written, the channel displays the Registration Code to the user along with instructions for the user to browse to the channel's registration web site.
5. Logic on the registration web site is responsible for creating the Linking Response file in the S3 Linking bucket after the customer successfully registers.
6. The channel checks the Linking bucket every retry-interval seconds (typically 30) for a Linking Response file with a name built from the Partner ID string and the Device Token.

### Example Linking Response File

File name: mycorp/ 9UR4TE54FWFBMFHB

```
<?xml version="1.0" ?>
<linkResponse>
  <status>success</status>
  <messageTitle>Message from Registration Server</messageTitle>
  <message>This message is displayed on the Roku player.</message>
  <deviceID>K0A09T000000</deviceID>
  <customerID>0123456789</customerID>
  <creationTime>2012-06-23T23:18:20.000Z</creationTime>
  <expirationMinutes>43200</expirationMinutes>
</linkResponse>
```

## Device Linking

1. If the Linking file is present, the channel retrieves it. If the file is not found and read within retry-duration seconds (typically 900) the registration and linking process fails and the user is notified with an implementation dependent error message.
2. If the Linking file contains a command it is executed. Currently the only command implemented is "deleteToken" which is used to remotely erase the device token.

### Example Linking Response File with Command

```
<?xml version="1.0" ?>
<linkResponse>
  <status>failure</status>
  <messageTitle>Message from Registration Server</messageTitle>
  <message>Your subscription has been cancelled.</message>
  <command>deleteToken</command>
  <deviceID>K0A09T000000</deviceID>
  <customerID>0123456789</customerID>
  <creationTime>2012-06-23T23:20:02.000Z</creationTime>
  <expirationMinutes>43200</expirationMinutes>
</linkResponse>
```

3. If the Linking Response file contains a message it is displayed to the user.
4. If the status flag contained in the Linking file is "success", the channel stores the Device Token into the player's registry and channel operation continues normally. The player is now registered and linked to the channel.
5. If the status element in the Linking Response file is not "success" then the registration and linking process fails and access to the channel or portions of the channel protected by registration and linking is blocked.
6. If the status element in the Linking Response file is not "success" and no "message" element is present, an implementation dependent error message is displayed to the user.
7. The channel does not attempt to verify or otherwise make use of the device ID or customer ID in the Linking Response file.

## Post-Registration

1. If the channel has a Device Token stored in the registry, it will use that token along with the Partner ID to form a Linking Response file name to fetch from the Linking bucket.
2. If the Linking Reponse file is present, the channel retrieves it. If the file cannot be retrieved an implementation dependent error message is displayed to the user.
3. Processing continues per **Device Linking** step #2 above, with the exception that after the optional message is displayed, "expirationMinutes" and "creaionTime" elements are compared to the current time to determine if the registration has expired.
4. If the registration has expired an optional expiration message is displayed, and the player starts over at step #1 of the Pre-Registration phase above.
5. The channel has no mechanism available for the user to delete the Device Token. The token may only be deleted by a <deleteToken> command in the Linking Response file, by re-entering and completing the Pre-Registration and Device Linking phases after a registration token has expired, or if the user uninstalls the channel.
6. A channel can be remotely disabled on a user's player by three different methods:
  - a. The status element in the Linking Response file can be set to "failure". It is recommended that the Linking Response file also include a message to be displayed to the user explaining why the channel has been disabled. When the portion of the channel protected by registration and linking is accessed the channel will display the message contained in the Linking Response file. If no message is contained in the Linking Response file the channel will display an implementation dependent error message.
  - b. The <deleteToken> command can be placed in the Linking Response file. It is recommended that the Linking Response file also include a message to be displayed to the user explaining why the channel has been disabled. When the portion of the channel protected by registration and linking is accessed the channel will display the message contained in the Linking Response file, followed by an optional implementation dependent error message.
  - c. The Linking Response file can be removed from the Linking bucket. When the portion of the channel protected by registration and linking is accessed the channel will display an implementation dependent error message. The channel will need to be uninstalled then re-installed in order to acquire a new token.