PlayReady 3.0 User Guide

Version 0.9

Broadcom
1320 Ridder Park Drive
San Jose, California 95131
broadcom.com

Table of Contents

1 Revision History

Issue	Date	By	Change
0.9	3/29/16	Hugo Saint-Laurent	Initial Draft

2 Introduction

This document outlines how to build and run the PlayReady 3.0 software bundle on Broadcom's set-top boxes. The target audience is developers familiar with the Playready SDK that want to run Broadcom's Playready stack on their board.

This document does not provide information on the Playready 3.0 architecture. Refer to Microsoft Documentation for detailed documentation on that topic.

3 Reference

- Broadcom DRM Utility 4.x Application Note
- Microsoft Playready Documentation Pack version 3.0.1

4 Glossary

Abbreviation	Description
CRR	Compressed Restricted Memory Region
DRM	Digital Right Management
REE	Rich Execution Environment, used to describe the untrusted application environment in the system.
SAGE	Secure Applications Guardian Engine
SRAI	SAGE Remote Application Interface (SRAI)
SVP	Secure Video Path
URR	Uncompressed Restricted memory Region
TEE	Trusted Execution Environment, secure environment in which assets are protected, sensitive operations occur, etc.

5 Model Certificate Generation

A model certificate is needed per device model on which you support PlayReady. As mentioned Microsoft documentation Pack, the following steps are needed to create model certificate.

1. Create an unsigned template file for your device model.

2. Generate the PlayReady model certificate using Microsoft generatemodelcert.exe tool.

3. Generate a Playready DRM binary file using Broadcom's DRM Utility and place it on devices of the same model.

Please refer to MS Documentation Pack for further details.

5.1 Unsigned Template Creation

As a starting point, you can use Broadcom's reference template provided as part of URSR releases in BSEAV/lib/playready/3.0/template/Sage. And modify it based on your device needs.

The template must contain an UNSIGNEDTEMPLATE node and a FEATURE node.

5.1.1 UNSIGNEDTEMPALTE Node

The UNSIGNEDTEMPLATE node contains the manufacturer and device information in the following sub nodes

Node	Description
NAME	Device name.
MODEL	Model name or number.
SECURITYLEVEL	Integer specifying the security level of the device certificate. The default value is 150, which is used for test devices.
SECURITYVERSION	Optional node specifying the security version of the device certificate. Contains the <NUMBER> field that specifies the security version number. Broadcom TEE, the current security version is 0.0.0.1

5.1.2 FEATURE Node

The FEATURE node contains the sub node detailing which features are currently enabled on a device. It is store inside the UNSIGNEDTEMPLATE node

Field	Description

Field	Description
CLOCK	Tells what sort of clock is supported on the device for using time-based licenses. The following are the possible values: **0**—The device does not have a clock and cannot use time-based licenses. **1**—The device supports an anti-rollback clock. **2**—The device supports a secure clock.
PLAYREADY3FEATURES	0 or 1. Tells if the device supports PlayReady 3.0 and later features. Optional.
TRANSMITTER	0 or 1. Tells if the device supports being a transmitter for the network device streaming protocol. Optional.
RECEIVER	0 or 1. Tells if the device supports being a receiver for the network device streaming protocol. Optional.
SUPPORT_REVOCATION	0 or 1. Tells if the device supports certificate revocation lists

Here's an example of an unsigned template.

```
<UNSIGNEDTEMPLATE>
 <NAME>MyDeviceName</NAME>
 <MODEL>XR-700</MODEL>
 <SECURITYLEVEL>3000</SECURITYLEVEL>
 <SECURITYVERSION>
  <NUMBER>0.0.0.1</NUMBER>
 </SECURITYVERSION>
 <FEATURES>
  <CLOCK>2</CLOCK>
  <SUPPORT_REVOCATION>1</SUPPORT_REVOCATION>
  <TRANSMITTER>1</TRANSMITTER>
```

```
    <RECEIVER>1</RECEIVER>
    <PLAYREADY3FEATURES>1</PLAYREADY3FEATURES>
  </FEATURES>
</UNSIGNEDTEMPLATE>
```

In this example, the security level is set to 3000, and the security version to 0.0.0.1. Device models based on this template will be Playready 3.0 devices. They will use secure clock, support revocation, and have transmitter and receiver capabilities.

5.2 generatemodelcert.exe

The PlayReady model certificate is generated using Microsoft generatemodelcert.exe tool. Refer to MS porting kit documentation for details. This tool is provided as part of Microsoft Porting kit. Contact Microsoft for details.

The following example illustrates how to generate a Playready model certificate with SL3000 keys.

```
generatemodelcert.exe -z:PR -b:PRDACResponse3000.dat -f:PrivateKeyFile_SL3000.xml -
u:BRCM_Playready_Unsigned_Device_Template_SL3000.dat -g:PlayReadyModelCert3000.dat -
h:ModelCertPrivKeysPlayReady3000.dat
```

5.3 Playready Binary File Generation

Generate a Playready DRM binary file using Broadcom's DRM Utility and place it on devices of the same model. Broadcom's DRM Utility is not delivered as part of URSR releases. Please contact your Broadcom FAE to request it. Refer to the DRM Utility documentation for details.

6 Generalities

The following exports are used by Broadcom build system and should be defined.

1. `$URSR_TOP` is the path of URSR root directory.

   You can set it by

   Either,

   ```
   export URSR_TOP=<The_Top_directory_of_URSR_Source_code>
   ```

   Or if your current directory is the top directory of URSR source code, you can simply do:

   ```
   export URSR_TOP=$PWD
   ```

2. $NEXUS_BIN_DIR is the path of the output obj and bin files of the build. If you want to change the default path, do :

```
export NEXUS_BIN_DIR=<Path of the obj/bin files>

mkdir -p $NEXUS_BIN_DIR
```

Playready 3.0 only works with SAGE.

7 Enabling Playready 3.0 in Nexus

The following exports must be set to build Nexus with PlayReady support.

```
export MSDRM_PRDY_SUPPORT=y
export NEXUS_COMMON_CRYPTO_SUPPORT=y
```

Where:

- `MSDRM_PRDY_SUPPORT=y`: enables PlayReady support.

8 Building the PlayReady SDK library

This section is only pertinent to the developers with access to the PlayReady SDK source code.

You can skip this entire section if BSEAV/lib/playready/3.0 /source does not exist.

The following exports must be set to build Playready SDK source code.

```
export PRDY_TOP=<The_Top_directory_of_URSR_Source_code>
```

Where:

- `PRDY_TOP`: Path of Playready SDK root directory

8.1 Building PlayReady SAGE

To build the library with SAGE support, set the flags as follows

```
export PLAYREADY_HOST_BUILD=y
```

Then run "**`make clean all`**" from `${PRDY_TOP}/3.0/source`, followed by

"**`make clean all`**" from `${PRDY_TOP}/3.0/source/linux/libraries`

9 Device Certificates

When a PlayReady application runs for the first time, it will use the model certificate to create:

1) The data store file is created: datastore3x.hds
2) The device certificates, signing key pair and encryption key pair which are then stored in the data store.
3) The key history is stored in the file name: keyhistory3x.dat

These files are not compatible with the same files created by other versions of the PlayReady SDK. So you need to erase them if you change SDK versions.

Run the following commands to erase them from the directory where the PlayReady application ran.

```
rm *.dat
rm *.hds
```

10 DRM Rootfs Library

The DRM Rootfs library is used to specify the default paths to DRM assets on the file system. Customers have access to source code of this library. So they can change the default paths to fit their needs. All they have to do is modify the values in BSEAV\lib\drmrootfs\config_inc\drm_playready.inc and recompile the library.

To build this library, do the following steps

```
cd ${URSR_TOP}/BSEAV/lib/drmrootfs

unset NEXUS_MODE

make clean

make install
```

The new library will be copied to ${URSR_TOP}/BSEAV/lib/drmrootfs/lib/arm/linuxuser/ by the build system

11 PlayReady Examples

Currently 2 examples are available:

- prdy30_svp: Plays a stream on a secure video path using NxClient/NxServer

- prdy30_svp_single: Play a stream on a secure video path using Nexus compiled in single process mode.

11.1 Application usage

⚠ **Caution!** The following two applications assume that the second parameter on the command line is the file to play. Placing the –vc1 or –secure parameter before the file will cause the application to fails.

11.1.1.1 prdy30_svp_single

Name	prdy30_svp_single: Plays a PlayReady protected stream using nexus compiled in single process mode.
Synopsis	prdy30_svp_single <input file> [option(s)]
Parameters	`-vc1`: Optional parameter used to specifies that the stream to play is vc1 encoded. If this parameter is not set, the application will assume that the stream is encoded in H264 format.

11.1.1.2 prdy30_svp

Name	prdy30_svp: Plays a PlayReady protected stream using NxClient/NxServer.
Synopsis	prdy30_svp <input file> [option(s)]
Parameters	`-vc1`: Optional parameter used to specifies that the stream to play is vc1 encoded. If this parameter is not set, the application will assume that the stream is encoded in H264 format.

11.2 Build PlayReady Secure Video Path Examples

11.2.1 Common Environment Setup

Please adjust the build tools to your specific URSR release. This may include Linux and stbgcc toolchain.

1) Go to the root folder of your URSR sources. For example:

```
cd $URSR_TOP
```

2) Prepare the environment variables of your board. Please refer to the URSR document for the exports of a particular reference board.

```
export NEXUS_PLATFORM=<Platform_ID>
export BCHP_VER=<Revision>
export LINUX=<Kernel_path>
export TOOLCHAIN=<Toolchain_path>
export PATH=$TOOLCHAIN:$PATH
# arm-linux for ARM-based cores and mipsel-linux for MIPS-based cores.
export B_REFSW_ARCH=<ARM>

#Below is needed only if your board has a subtype like SV, VMS_SFF, C,
DBS….etc.

export NEXUS_USE_<ChipID>_<BoardSubType>=y
export B_REFSW_ARCH=arm-linux
```

For a 97252 D0 board which uses Linux 3.14-1.8 and is compiled with stbgcc-4.8-1.2 tool chain, the environment variables would be set as follows:

```
export NEXUS_PLATFORM=97252
export BCHP_VER=D0
export LINUX=/<Path>/linux-3.14-1.8/7250b0
export TOOLCHAIN=/<Path>/stbgcc-4.8-1.2/binexport
PATH=$TOOLCHAIN:$PATH
```

Export the following environment variables:

```
export NEXUS_MODE=proxy
export NEXUS_COMMON_CRYPTO_SUPPORT=y
export MSDRM_PRDY_SUPPORT=y
export SAGE_SUPPORT=y/n
export BMRC_ALLOW_XPT_TO_ACCESS_KERNEL=y (needs to be set to y when
SAGE_SUPPORT=y)
export NEXUS_HDCP_SUPPORT=y/n (needs to be set to y when SAGE_SUPPORT=y)
```

Note that Nexus build system will set SAGE_SUPPORT to ‘y’ if it’s not explicitly specified. This is not the case for common drm makefiles.

⚠ **Caution!** The test stream must be encrypted, these applications can’t play clear format.

11.2.2 Execution Environment Setup

1) Copy the playready3x.bin into obj.<platform>/nexus/bin/.

   If you have built the application with SAGE support enabled. Then you must use the playready3x.bin that is bound for your chip.

2) Change permissions of obj.<platform>/nexus/bin/ as follows:

```
chmod –R 777 obj.<platform>/nexus/bin/
```

3) Mounting preferences for a view can vary from user to user depending on requirements. Below is an example to mount obj.<platform>/nexus/ directory to /mnt/nfs/ with read and write permissions

```
mount -t nfs –o nolock,rw  <IPaddr_or_hostName>:/<Path of
URSR>/obj.97445/nexus/bin /mnt/nfs
```

11.2.3 Compilation

1) Build the examples:

```
cd ${URSR_TOP}/BSEAV/lib/playready/3.0/examples; make target=prdy30_svp
```

The example build system will build all the required libraries.

If you define NEXUS_BIN_DIR variable, the executables will be placed under the path defined in NEXUS_BIN_DIR. Otherwise they will be placed under obj.<platform>/nexus/bin/.

11.2.4 Run prdy30_svp

1) Go into the Nexus binary folder and start NxSever:

```
cd /mnt/nfs

./nexus nxserver –svp_urr &
```

Wait about 10 seconds for the server to start.

2) Launch the test application:

```
./nexus.client prdy30_svp <videoName>.ismv &
```

If it runs properly, you will see the stream playback on the TV. On the other hand, if it fails, usually it's because your playready3x.bin is not right.

11.2.5 Run prdy30_svp_single

1) Go into the Nexus binary folder and start the application with the following parameters:

```
cd /mnt/nfs

./nexus prdy30_svp_single -preload <videoName>.ismv
```

If it runs properly, you will see the stream playback on the TV. On the other hand, if it fails, usually it's because your playready3x.bin is not right.

12 Troubleshooting PlayReady Problems

12.1 Playready3x.bin

1) Make sure you use a playready3x.bin that is bound to your chip. Take extra care if a socket board is used. Double check that the playready3x.bin file you got for a given OTP ID matches the OTP ID of the chip currently socketed in your board.

12.2 Common SDK errors

12.2.1 Drm_Initialize: Exiting with error [0x8003006E].

The error 0x8003006E is a file open error. This error will be reported if the PlayReady SDK fails to open any of the files mentioned above. The most common causes for this are as follows:

- Playready3x.bin has not been copied into obj.<chip_nb>/nexus/bin folder
- The permissions of obj.<chip_nb>/nexus/bin are not set to writable.
- A loadable TA is missing or doesn't have rw permission set on it.
- Starting Playready_svp or playready_svp_single and specifying –secure or –vc1 before the file to play will cause this error. Make sure you follow this semantic when `application <input file> [option(s)]`
- The target board wasn't mounted with –nolock,rw. Please use the following mount command:

```
mount -t nfs -o nolock,rw <your_mount_server>:<your_path>/obj.<chip_nb>/nexus/bin /mnt/nfs
```

12.2.2 Drm_Initialize: Exiting with error [0x8004C803]

The error 0x8004C803 means that an invalid feature entry was encountered OR the porting kit was linked with mutually incompatible features or features incompatible with the certificate. In general, customers use our precompiled libraries which rules out linking errors. However, they often include incompatible features in their device certificate template, like including both secure clock and anti-rollback support. Refer them to the golden certificate templates.

12.2.1 Drm_Initialize: Exiting with error [0x8004C805]

The error 0x8004C805 means that an invalid box security version was encountered. This usually means the security version used in the unsigned model certificate template doesn't match Broadcom's Playready TEE security version. Changes the value in the unsigned template and regenerate the model certificate and playready3x.bin.

12.2.2 Drm_Reader_Bind: Exiting with errors [0x8004C013]

The error 0x8004C013 means that a license wasn't found to play the requested content. The binding operation will fail with this error when an application is trying to play SL3000 content using keys with a lower security level like 150 or 2000.

12.2.1 Drm_Reader_Bind: Exiting with errors [0x80070057]

The error 0x80070057 means that an invalid argument has been received. Besides the obvious reason, this error is also return when an application tries to play a SL3000 stream without setting the decryption mode to OEM_TEE_AES128CTR_DECRYPTION_MODE_HANDLE. Double check that your application is calling Drm_Content_SetProperty() with the proper value.

12.2.2 Drm_Reader_DecryptOpaque failures

1) Make sure the buffers passed to Drm_Reader_DecryptOpque() have been allocated on a Nexus heap. Otherwise the virtual to physical address conversion that is done internally to setup the DMA transfer will fail.

2) When playing a protected stream using a secure video path, it's not enough to allocate buffers on any Nexus heap. Buffers MUST be allocated on Nexus's SECURE heap. Refer to the SVP Application Note for details.

12.2.1 All other PlayReady SDK error codes

All the errors codes returned by PlayReady are in ${URSR_TOP}/BSEAV/lib/playready/3.0/inc.drmresults.h. Please refer to this file for description and clues about what could be the issue.