



Remote DVR Access API Application Guide

Overview

Introduction

The Remote DVR Access APIs allow applications to access data managed by the SARA DVR recording system. The Remote DVR Access feature provides the following capabilities:

- Ability to retrieve the present recordings on the DVR.
- Ability to delete recordings on the DVR.
- Ability to retrieve the presently scheduled recordings.
- Ability to cancel scheduled recordings.
- Schedule future recordings including all-episode requests.
- Determine conflicts so that the user may be presented with the option of which recordings to cancel to resolve the request at the time of the scheduling or at a later time.
- Ability to retrieve the hard disk drive usage percentage.
- Ability to initiate playback of recordings.

To help ensure the consistency of the recording data, the following restrictions are enforced:

- Requests to modify the scheduled recording list are rejected while conflict resolution is in progress. Presently, the conflict resolution will timeout (if the user does not correct the conflict) after three minutes and remove the most recent request.
- If the local user (at the DVR) is editing data for a program and the remote user deletes the program, the local UI will exit the edit screen and return to the appropriate DVR list (either Scheduled List or Recorded List).
- The application will not be allowed to schedule a recording within three minutes of the program start time so that conflicts can be handled before the recording begins.

Purpose

This document explains the APIs provided by the SARA DVR recording system, so that application developers can create an application for users to manage their DVR remotely.

Scope

This document presents the APIs available to develop an application to access the SARA DVR information. There is no user authentication or security provided by these APIs.

In addition, the act of enabling DVR recording management over the network implies a load on the network capabilities of the DBDS network. This document does not address the network impact analysis, because the network architecture will vary for each service provider.

A guiding principle is to transfer the minimal amount of data across the network to reduce the bandwidth usage for remote DVR management applications.

In addition, this document makes the following assumptions:

- Functionality is for DVR set-tops (STBs) only.
- IPG specifics do not need to be sent across the network, because the web application has access to IPG information. However, the web application may occasionally need to query information for a single program at a time.
- Scheduling capabilities are the same as provided by the SARA UI on a DVR box. For example, padding of start and end times is not allowed.
- Data exchanged between SARA and the Remote DVR application will use pointers instead of handles.

Audience

This document contains useful information for application developers who create applications for the DBDS.

Document Version

This is the second formal release of this document.

API Description

All of the APIs provided are asynchronous in nature so that the calling application is not blocked while the requested operation is being performed.

Program Definition

A **program** is defined with the following data structure:

```

Data      struct dvrSchedule_t {
Structure  int          dcn;           // Channel number
              time_t      start;        // UNIX epoch
              time_t      duration;     // duration in seconds
              int         saveDays;
              int         saveEpisodes;
              int         allEpisodeDefinition;
              struct {
                uil6      repeat : 1;    // repeat flag
                uil6      isManual   : 1;
                // manual recording flag
                uil6      isPlayback  : 1;
                // presently in playback (only set on query)
                uil6      isRecording : 1;
                // presently recording (only set on query)
                uil6      reserved   : 12;
              } flags;
            } ;

```

Parameters	<i>dcn</i>	Channel number (range from 0-1999)
	<i>saveDays</i>	The number of days that an episode will be saved. Presently, SARA supports the following values: <ul style="list-style-type: none"> ■ 0 = Keep Until User Erases ■ 1 = Keep 1 Day ■ 2 = Keep 2 Days ■ 7 = Keep 7 Days ■ 14 = Keep 14 Days ■ Any other value is an error at this time
	<i>saveEpisodes</i>	The number of episodes in an all episode request to save. Presently, the following values are supported: <ul style="list-style-type: none"> ■ 0 = Keep all episodes ■ 1-5 = Keep the specified number of episodes ■ Any other value is an error at this time

- allEpisodeDefinition* The type of all episode request. The following values are defined at this time:
- 0 = all episodes on this channel
 - 1 = all episodes on all channels with this title (defined but not supported at this time)
 - 2 = all episodes on this channel at this time slot on *any* day of the week
 - 3 = all episodes on this channel which are first run
 - 4 = all episodes on this channel at this time slot on *this* day of the week (This option is not supported in DVR1.5.2 release.)
 - Any other value is an error at this time

Note: Use of the first run all episode request is dependent upon having the appropriate IPG data available *and* SAM service definition.

IPG Data Definition

The **IPG data for Scheduled or Recorded Programs** is represented with the following:

Data Structure	<pre> struct dvrProgramData_t { char title[60]; // Program Title (null terminated) char longDescription[250]; // Long Description (null terminated) ui32 logoId; // Channel Logo Id (ie. HBO, STARZ, ..) time_t lifetime; // Recorded Program only (see notes) struct { ui16 halfStar : 1; // ½ star flag ui16 wholeStars : 3; // # of whole stars ui16 reserved1 : 4; ui16 rating : 4; // rating ui16 reserved2 : 4; } description; // from epg2 struct { ui16 cc : 1; // closed caption flag ui16 stereo : 1; // stereo flag ui16 surround : 1; // surround flag ui16 sap : 1; // SAP flag ui16 reserved : 12; } flags; // from epg1 // the following will be added if possible struct { ui16 firstRun : 1; // first run (Scheduled only) ui16 HD : 1; // HD program (TBD) ui16 reserved : 14; } additionalInfo; ui8 stoppedEarly; }; </pre>						
Parameters	<table border="0"> <tr> <td style="padding-right: 20px;"><i>lifetime</i></td> <td>For non-episodic recorded programs, the time at which the program will be deleted. Using UNIX epoch.</td> </tr> <tr> <td><i>halfStar</i></td> <td>(1 = half star, 0 = NO half star)</td> </tr> <tr> <td><i>wholeStars</i></td> <td>Number of whole stars for program critique.</td> </tr> </table>	<i>lifetime</i>	For non-episodic recorded programs, the time at which the program will be deleted. Using UNIX epoch.	<i>halfStar</i>	(1 = half star, 0 = NO half star)	<i>wholeStars</i>	Number of whole stars for program critique.
<i>lifetime</i>	For non-episodic recorded programs, the time at which the program will be deleted. Using UNIX epoch.						
<i>halfStar</i>	(1 = half star, 0 = NO half star)						
<i>wholeStars</i>	Number of whole stars for program critique.						

<i>rating</i>	<p>Parental rating for the program. Presently, SARA supports the following values:</p> <ul style="list-style-type: none">■ 0 = Unknown■ 1 = NR■ 2 = TV-Y■ 3 = TV-Y7■ 4 = G/TV-G■ 5 = PG/TV-PG■ 6 = PG13■ 7 = TV-14■ 8 = TV-MA■ 9 = R■ 10 = NC17■ 11 = NR-Adult■ 12-15 are reserved for future use
<i>stoppedEarly</i>	<p>Specifies why a recording stopped. Presently, SARA supports the following values:</p> <ul style="list-style-type: none">■ 0 = Normal stoppage■ 1 = Out of Disk Space■ 2 = User Initiated■ 3 = SwitchedDigital Service Unavailable■ 4 = SwitchedDigital Service Redirected■ 5 = PatPmtChange■ 6 = EAS Force Tune activation■ 7 = Settop reboot■ 8 = CableCARD™ Software Upgrade initiated■ 9-255 are reserved for future use

List of Programs Definition

Data struct dvrScheduleList_t {
Structure ui16 numberOfPrograms;
 dvrSchedule_t program[];
 // actually variable length depending upon
 // numberOfPrograms.
 } ;

Parameters None

Type of List Definition

A **type of list** is described with the following:

```

Data      typedef enum {
Structure     eRecordedPrograms = 0,
                 eScheduledPrograms,
                 eConflictingPrograms
                 } eDvrListType;

// There will be no PTV error codes returned. Any errors that
// occur internally are to be translated into one of the
// following (add new ones as required):
#define kSaraDvr_NoErr          0x00000000

#define kSaraDvr_InvalidTimeoutErr    -0x01
// an unsupported timeout was specified
// in the Ptv_Reply structure

#define kSaraDvr_InvalidListErr       -0x02
// the specified list type is unknown or
// unsupported by the API.

#define kSaraDvr_EmptyListErr         -0x03
// the list is empty

#define kSaraDvr_TimeBufferErr        -0x04
// the required advanced programming
// time has not been provided.

#define kSaraDvr_InvalidProgramErr    -0x05
// invalid information in the
// specified program

#define kSaraDvr_InvalidQueueErr      -0x06
// no queue specified in Ptv_Reply
// structure

#define kSaraDvr_ConflictErr          -0x07
// a scheduling conflict exists.

#define kSaraDvr_NoProgramsErr        -0x08
// no programs exist for the list.

#define kSaraDvr_MemoryErr            -0x09
// an error occurred while allocating
// memory

#define kSaraDvr_PartialDeletionErr   -0x0a
// only a subset of the deletion request
// completed successfully.

#define kSaraDvr_UnAuthorizedErr      -0x0b
// DVR is not authorized on the STB.

#define kSaraDvr_DiskFullErr          -0x0c
//DVR HardDisk is full and no further
//scheduling may take place

#define kSaraDvr_MaxRecordingReqErr    -0x0d
// Maximum number of scheduled
// recordings is exceeded.

#define kSaraDvr_ConflictInProgressErr -0x0e
// Conflict Resolution is in progress and
// must complete before performing
// scheduling operations.

#define kSaraDvr_NotReadyErr          -0x0f
// DVR is not ready (still initializing).

#define kSaraDvr_AlreadyScheduledErr  -0x10
// the requested program is already
// scheduled.

```

Type of List Definition

```
#define kSaraDvr_PpvNotPurchasedErr      -0x11
// the PPV program being scheduled for
// recording has not been purchased.

#define kSaraDvr_InvalidParameterErr    -0x12
// Invalid parameter was specified

#define kSaraDvr_NoPlaybackErr          -0x13
// Unable to activate the playback.

#define kSaraDvr_NotSupportedErr        -0x14
//The API is not supported.

#define kSaraDvr_NoProgramsScheduledErr -0x15
//No programs were scheduled with the
//request.
```

Parameters None

SARA DVR API Responses

All of the SARA DVR APIs are asynchronous in nature. The response event is generically defined in the following manner:

```
kDt_SaraDvr | kEt_AsyncResponse
```

where

- x = response identifier (rid) specified by the caller in the Ptv_Reply structure.
- y = response data which is specific to the API executed.
- z = return code if an error occurred (zero is returned if no error occurred).

Scheduling Change Event

The following event is generated when a scheduling change has occurred so that the application or local UI may update appropriately:

```
kDt_SaraDvr | kEt_ListUpdate
```

where

- $x = \text{eDvrListType}$.
- $y = \text{unused}$.
- $z = \text{unused}$.

Schedule Recordings Remotely

The following API is used to schedule recordings remotely. The caller is expected to free the conflict list after handling any conflicts.

Syntax	<code>extern int saraDvr_AddFutureRecording (dvrSchedule_t *program, Ptv_Reply *reply);</code>	
Parameters	<i>program</i>	Pointer to the structure defining the program for which a recording is to be scheduled.
	<i>reply</i>	Pointer to a Ptv_Reply structure.
	Note: The reply→responseQ must be specified and reply→timeout must be kPtv_Forever.	
Returns	<i>kSaraDvr_NoErr</i>	No error. A kDt_SaraDvr kEt_AsyncResponse event will be sent.
	<i>kSaraDvr_InvalidTimeoutErr</i>	Unsupported timeout value in reply->timeout field.
	<i>kSaraDvr_InvalidProgramErr</i>	The parameters for the specified program are incorrect including any attempt to record a channel that is unsupported (like VOD, PVRAC, etc.).
	<i>kSaraDvr_TimeBufferErr</i>	Program is being scheduled for recording too close to the start time of the recording.
	<i>kSaraDvr_InvalidQueueErr</i>	No responseQ specified in reply structure.
	<i>kSaraDvr_UnauthorizedErr</i>	DVR is not authorized on the STB.
	<i>kSaraDvr_NotReadyErr</i>	DVR is not initialized.
	<i>kSaraDvr_NotSupportedErr</i>	This API is not supported on the platform.
Comments	The kDt_SaraDvr kEt_AsyncResponse event sent in response to the request has the following information:	
	<ul style="list-style-type: none"> ■ x = The response identifier specified in the original Ptv_Reply structure. ■ y = Pointer to a dvrScheduleList_t structure which describes the programs in conflict. The recipient of the event must free this structure. ■ z = Result of request. 	
	<i>kSaraDvr_NoErr</i>	When scheduling completes successfully with no errors or conflicts.
	<i>kSaraDvr_ConflictErr</i>	A scheduling conflict has occurred and the y field points to a list of conflicting programs.
	<i>kSaraDvr_MemoryErr</i>	A memory allocation failure has occurred. The caller should attempt to check for scheduling conflicts. The y field will be set to NULL.

Schedule Recordings Remotely

<i>kSaraDvr_MaxRecordingReqErr</i>	Maximum number of scheduled recordings is exceeded. User needs to delete recording requests to add new requests.
<i>kSaraDvr_ConflictInProgressErr</i>	Conflict resolution is in progress and must complete before new scheduling requests may be attempted.
<i>kSaraDvr_DiskFullErr</i>	Minimum amount of disk space is no longer available. The y field will be set to NULL.
<i>kSaraDvr_AlreadyScheduledErr</i>	The requested program is already scheduled for recording. The y field will be set to NULL.
<i>kSaraDvr_PpvNotPurchasedErr</i>	The requested program to be recorded is a PPV event and has not been purchased. The y field will be set to NULL.
<i>kSaraDvr_NoProgramsScheduledErr</i>	No programs were scheduled for recording based on the request submitted via the API.

Retrieve List

The following API is used to retrieve either the Scheduled, Recorded, or Conflict list. The caller is expected to free the returned list.

Note: The retrieval of the Conflict List may not be available in the initial engineering release.

Syntax `extern int saraDvr_GetList (eDvrListType listType, Ptv_Reply *reply);`

Parameters *listType* Parameter specifies the type of list to be retrieved.

Note: Specifying eDvr_ConflictingPrograms will return a list of all conflicting programs. The caller is expected to determine which programs are conflicting with each other.

reply Pointer to a Ptv_Reply structure.

Note: The reply→responseQ must be specified and reply→timeout must be kPtv_Forever.

Returns	<i>kSaraDvr_NoErr</i>	No error. A kDt_SaraDvr kEt_AsyncResponse event will be sent.
	<i>kSaraDvr_InvalidTimeoutErr</i>	Unsupported timeout value in reply→timeout field.
	<i>kSaraDvr_InvalidListErr</i>	The specified listType does not exist.
	<i>kSaraDvr_InvalidQueueErr</i>	No responseQ specified in reply structure.
	<i>kSaraDvr_UnauthorizedErr</i>	DVR is not authorized on the STB.
	<i>kSaraDvr_NotReadyErr</i>	DVR is not initialized.
	<i>kSaraDvr_NotSupportedErr</i>	This API is not supported on the platform.

Comments The kDt_SaraDvr | kEt_AsyncResponse event sent in response to the request has the following information:

- x = The response identifier specified in the original Ptv_Reply structure.
- y = Pointer to a dvrScheduleList_t structure which describes the programs in the requested list. The recipient of the event must free this structure.
- z = Result of request.

kSaraDvr_NoErr The requested program list has been successfully retrieved.

kSaraDvr_NoProgramsErr There are no programs in the requested list. The y field is set to NULL.

kSaraDvr_MemoryErr A memory allocation failure has occurred. The y field is set to NULL.

Retrieve List

Note: The response when retrieving the Recorded List will default any unused fields. Presently, this means saveDays, saveEpisodes, allEpisodeDefinition, repeat, and isManual fields of dvrSchedule_t will be returned with default values of zero (0).

Delete Items

The following API is used to delete items from either the Scheduled or Recorded list. The list will NOT be freed by the API.

Syntax

```
extern int saraDvr_DeleteList ( eDvrListType listType,
dvrScheduleList_t *changeList, Boolean force, Ptv_Reply *reply
);
```

Parameters

<i>listType</i>	Parameter specifies the type of list to be managed.
<i>changeList</i>	List of programs to be deleted. The caller must not touch this list after calling this API until either an error is returned by the API or the response event is received.
<i>force</i>	Normally, the API will not delete Scheduled Programs that are presently recording. This option will override this protection. It will <i>not</i> delete a program that is presently being played back by the local user. Only <i>one</i> program may be specified in the <i>changeList</i> when this parameter is set to TRUE and it is only valid for the Scheduled List.
<i>reply</i>	Pointer to a Ptv_Reply structure. Note: the reply→responseQ must be specified and reply→timeout must be kPtv_Forever.

Returns

<i>kSaraDvr_NoErr</i>	No error. A kDt_SaraDvr kEt_AsyncResponse event will be sent.
<i>kSaraDvr_InvalidTimeoutErr</i>	Unsupported timeout value in reply→timeout field.
<i>kSaraDvr_InvalidProgramErr</i>	More than one program specified with the force option.
<i>kSaraDvr_InvalidListErr</i>	The specified listType is not supported by this API.
<i>kSaraDvr_InvalidQueueErr</i>	No responseQ specified in reply structure.
<i>kSaraDvr_UnauthorizedErr</i>	DVR is not authorized on the STB.
<i>kSaraDvr_NotReadyErr</i>	DVR is not initialized.
<i>kSaraDvr_NotSupportedErr</i>	This API is not supported on the platform.

Comments The kDt_SaraDvr | kEt_AsyncResponse event sent in response to the request has the following information:

- x = The response identifier specified in the original Ptv_Reply structure.
- y = Not used.
- z = Result of request.

<i>kSaraDvr_NoErr</i>	The requested program list has been successfully changed.
-----------------------	---

Delete Items

<i>kSaraDvr_NoProgramsErr</i>	No programs remain in the list after the completion of the request.
<i>kSaraDvr_PartialDeletionErr</i>	Only a subset of the request completed successfully. One reason for receiving this error is that a program is being viewed and at the same time was put into the changeList.
<i>kSaraDvr_ConflictInProgressErr</i>	Conflict resolution is in progress and must complete before deletion of scheduled recordings may be attempted (only for eDvr_ScheduledPrograms).

Retrieve IPG Data

The following API is used to retrieve IPG data for a particular program from either the Scheduled or Recorded list. The storage for the data must be pre-allocated by the caller.

Syntax `extern int saraDvr_RetrieveProgramData (eDvrListType listType, dvrSchedule_t *program, dvrProgramData_t *programData, Ptv_Reply *reply);`

Parameters

<i>listType</i>	List from which the program data is to be retrieved.
<i>program</i>	Pointer to dvrSchedule_t. This parameter is filled out by the caller to allow the API to find the program for which the data is to be returned. The caller must not touch this buffer after calling this API until either an error is returned by the API or the response event is received.
<i>programData</i>	Pointer to dvrProgramData_t. The caller must allocate this buffer and not touch this buffer after calling this API until either an error is returned by the API or the response event is received.
<i>reply</i>	Pointer to a Ptv_Reply structure.

Note: The reply→responseQ must be specified and reply→timeout must be kPtv_Forever.

Returns

<i>kSaraDvr_NoErr</i>	No error. A kDt_SaraDvr kEt_AsyncResponse event will be sent.
<i>kSaraDvr_InvalidTimeoutErr</i>	Unsupported timeout value in reply→timeout field.
<i>kSaraDvr_InvalidQueueErr</i>	No responseQ specified in reply structure.
<i>kSaraDvr_InvalidProgramErr</i>	The specified program information is invalid.
<i>kSaraDvr_UnauthorizedErr</i>	DVR is not authorized on the STB.
<i>kSaraDvr_NotReadyErr</i>	DVR is not initialized.
<i>kSaraDvr_NotSupportedErr</i>	This API is not supported on the platform.

Comments The kDt_SaraDvr | kEt_AsyncResponse event sent in response to the request has the following information:

- x = The response identifier specified in the original Ptv_Reply structure.
- y = The pointer originally specified as the programData parameter is returned.
- z = Result of request.

<i>kSaraDvr_NoErr</i>	The requested program list has been successfully changed.
-----------------------	---

Retrieve IPG Data

kSaraDvr_InvalidProgramErr The requested program no longer exists in the specified list. This situation may occur due to the completion of a recording or deletion of a scheduled recording request or recorded program.

Initiate Playback

The following API is used to initiate a playback. Execution of this API allows one of the following:

- Playback recording immediately in main.
- Activate the Recorded Programs Option or Playback Status screen in main. If the recording has a bookmark (i.e., it has been played partially earlier), the Playback Status screen will be shown. Otherwise, the Recorded Programs Option screen will be displayed.

Note: Some playback options are not supported in the DVR 1.5.2 release.

```

Syntax      typedef enum eDvrRecordedPgmOptions
                {
                eDvr_PlayFromBeginning =0, // ePlay
                eDvr_PlayFromCurrentPosition, //This option is not supported
                                                //in the DVR 1.5.2 release.
                eDvr_SkipToEnd, //This option is not supported
                                                //in the DVR 1.5.2 release.
                eDvr_Erase,
                eDvr_PlayImmediately,
                // do not display the Recorded Program Options screen.
                eDvr_InvalidOption
                } eDvrRecordedPgmOptions;

                typedef enum eDvrDestination
                {
                eDvr_Main,
                eDvr_InvalidDestination
                } eDvrDestination;

                struct dvrDestination_t {
                eDvrDestination destination;
                // Add options necessary for various destinations later...
                };

                extern int saraDvr_RecordedProgramOptions (
                dvrSchedule_t *program,
                dvrDestination_t *displayDestination
                eDvrRecordedPgmOptions optionType,
                Ptv_Reply *reply);

```

Parameters	<i>program</i>	Pointer to dvrSchedule_t. This parameter is filled out by the caller to allow the API to find the program for which the data is to be returned. The caller must not touch this buffer after calling this API until either an error is returned by the API <i>or</i> the response event is received.
	<i>displayDestination</i>	Pointer to dvrDestination_t. This parameter is filled out by the caller to allow the API to determine the destination of the playback. Presently, only the destination of "Main" is supported. Specifying any other destination will result in the kSaraDvr_InvalidParameterErr being returned.
	<i>optionType</i>	Recorded Program Options which allows the caller to set the highlighted option.

Initiate Playback

reply Pointer to a Ptv_Reply structure.

Note: The *reply*→*responseQ* must be specified and *reply*→*timeout* must be *kPtv_Forever*.

Returns	<i>kSaraDvr_NoErr</i>	No error. A <i>kDt_SaraDvr</i> <i>kEt_AsyncResponse</i> event will be sent.
	<i>kSaraDvr_InvalidParameterErr</i>	Unsupported parameter specified by caller. Note: The DVR 1.5.2 release does not support <i>eDvr_PlayFromCurrentPosition</i> or <i>eDvr_SkipToEnd</i> . If the user selects either of these actions, the DVR will default to <i>eDvr_PlayFromBeginning</i> (<i>ePlay</i>) in the Recorded Program Option screen.
	<i>kSaraDvr_InvalidTimeoutErr</i>	Unsupported timeout value in <i>reply</i> → <i>timeout</i> field.
	<i>kSaraDvr_InvalidQueueErr</i>	No <i>responseQ</i> specified in <i>reply</i> structure.
	<i>kSaraDvr_UnAuthorizedErr</i>	DVR is not authorized on the STB.
	<i>kSaraDvr_NotReadyErr</i>	DVR is not initialized.
	<i>kSaraDvr_NotSupportedErr</i>	This API is not supported on the platform.

Comments The *kDt_SaraDvr* | *kEt_AsyncResponse* event sent in response to the request has the following information:

- *x* = The response identifier specified in the original *Ptv_Reply* structure.
- *y* = Not Used. Set to NULL.
- *z* = Result of request.

<i>kSaraDvr_NoErr</i>	The requested program has been successfully activated for the recorded program option screen.
<i>kSaraDvr_NoProgramsErr</i>	There are no programs in the requested list. Unable to locate the requested program.
<i>kSaraDvr_NoPlaybackErr</i>	Unable to do a playback.
<i>kSaraDvr_UnauthorizedErr</i>	DVR is not authorized on the STB.

Retrieve DVR-Related Data

The following API is used to allow retrieval DVR-related data.

Note: At present, this API only supports retrieval of HDD usage (in percentage). This API is not supported in the DVR 1.5.2 release.

Syntax

```
typedef enum eDvrGetName
{
    eDvr_HddUsagePercentage = 0,          // Percentage of HDD
                                          // used (0 - 100).
    eDvr_InvalidName
} eDvrGetName;

extern int saraDvr_Get ( eDvrGetName name, Ptv_Reply *reply);
```

Parameters

<i>name</i>	Parameter specifies the type of data requested.
<i>reply</i>	Pointer to a Ptv_Reply structure.

Note: The reply->responseQ must be specified and reply->timeout must be kPtv_Forever.

Returns

<i>kSaraDvr_NoErr</i>	No error. A kDt_SaraDvr kEt_AsyncResponse event will be sent.
<i>kSaraDvr_InvalidParameterErr</i>	Unsupported parameter specified by caller.
<i>kSaraDvr_InvalidTimeoutErr</i>	Unsupported timeout value in reply->timeout field.
<i>kSaraDvr_InvalidQueueErr</i>	No response specified in reply structure.
<i>kSaraDvr_UnauthorizedErr</i>	DVR is not authorized on the STB.
<i>kSaraDvr_NotReadyErr</i>	DVR is not initialized.
<i>kSaraDvr_NotSupportedErr</i>	This API is not supported on the platform.

Comments The kDt_SaraDvr | kEt_AsyncResponse event sent in response to the request has the following information:

- x = The response identifier specified in the original Ptv_Reply structure.
- y = Generally, a pointer to the requested data.
In the case of eDvr_HddUsagePercentage, the y field contains the data so additional memory does not need to be allocated.
- z = Result of request.

<i>kSaraDvr_NoErr</i>	The requested data has been successfully retrieved.
<i>kSaraDvr_InvalidParameterErr</i>	No data or an error in retrieving the data requested.
<i>kSaraDvr_NotReadyErr</i>	The DVR is not initialized.

Retrieve DVR-Related Data

For Information

If You Have Questions

If you have technical questions, call Cisco Services for assistance. Follow the menu options to speak with a service engineer.



Cisco Systems, Inc.
5030 Sugarloaf Parkway, Box 465447
Lawrenceville, GA 30042

678 277-1120
800 722-2009
www.cisco.com

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL:

www.cisco.com/go/trademarks

CableCARD is a trademark of CABLE Television Laboratories, Inc.

Other third party trademarks mentioned are the property of their respective owners.

The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Product and service availability are subject to change without notice.

© 2009, 2012 Cisco and/or its affiliates. All rights reserved.

September 2012 Printed in USA

Part Number 78-4029824-01 Rev B