SeaChange Interactive TV
Movies On Demand Service
For ITV Version 2.0

Authorization Check Plug-In Dll Specification

Revision 2.0
October 18, 2004

Paul Sullivan

SEACHANGE
INTERNATIONAL

124 ACTON STREET, MAYNARD, MASSACHUSETTS 01754

PHONE: 508-897-0100
FAX: 508-897-0132

Revision History

Revision	Date	Who	Description
1.0	November 28, 2000	pcs	Created
1.01	December 13, 2000	pcs	Renamed from MCC_PlugIn_specification.doc because the phrase ‘Authorization Check’ replaced ‘Credit Check.’
2.0	October 18, 2004	pcs	Modified and extended the interface.

Preface

This document specifies the programming interface for a Movies On Demand Service's Authorization Check Plug-In Dll.

Table of Contents

Content

REVISION HISTORY 3

PREFACE 4

TABLE OF CONTENTS.. 5

Content.. 5

Introduction 6

APPLICATION PROGRAMMING INTERFACE.. 7

API Details.. 7
General Information 7
Sequence of calls 7
Sample code 7
MOD Registry values for Plug-In Dll 8
MacInitialize 9
MacProcess 10
MacReport... 11
MacUninitialize 15

DATA STRUCTURES 16

Introduction

This section details the Movies On Demand Service Plug-In DLL architecture for interfacing with external systems. Prior to version 2.0, the DLL was only used for authorization checking. Version 1 of this interface was never used and is no longer supported.

The goal of this architecture is to provide a standard and extensible way for the Movies On Demand Service (MOD) to support multiple interfaces to external systems. This will be done through plug-in DLLs.

If a plug-in DLL is specified, MOD will load it at startup. Once it has been loaded, MOD invokes the **MacInitialize** function to allow that DLL to perform any initialization actions. For example, a DLL may need specific configuration information from the registry or may specify that a subset of data be supplied by MOD when invoking other entry points in the DLL.

After MOD has been informed of a request to play a movie, but before a stream starts playing, MOD can invoke an exported function that all plug-ins must support, specifically MacProcess. **MacProcess** is passed several session setup parameters and a subset of IDS data, in TLV format (type, length, value), specified by MacInitialize. The plug-in can perform whatever authorization checks are needed to either accept or reject the request.

The plug-in has the opportunity to modify several of the IDS data fields prior to returning control to MOD from its MacProcess function. Two of the three callbacks supplied to the plug-in via MacInitialize are used for this purpose. **pfnAllocRtn** is used to allocate a TLV buffer and **pfnUpdateRtn** is given the updated data in TLV format. MOD will update the data in the stream’s session object and free the memory allocated for the TLV.

MOD can invoke an exported function that a plug-in can optionally support, specifically MacReport. **MacReport** is passed a subset of IAD data, in TLV format, specified by MacInitialize. The same data used to update IAD’s billing information is available to the plug-in DLL when a stream is terminated. In addition, a stream termination code and a flag indicating a new purchase or a resume are available.

When MOD terminates, it calls the **MacUninitialize** function for a loaded plug-in DLL. This function gives the plug-in an opportunity to cleanly shutdown.

Application Programming Interface

API Details

General Information

The plug-in DLL consists of three required functions:

MacInitialize	Initialize
MacProcess	Process an authorization check of a request to stream a movie.
MacTerminate	Uninitialize

The plug-in DLL can optionally define the following function if it wants billing data reported to it when a stream is terminated.

MacReport	Process billing data for a terminated stream.

Sequence of calls

MOD will invoke MacInitialize at startup.

MOD will invoke MacProcess to determine whether or not a client is authorized to stream a movie. This happens after MOD has been informed of a request to play a movie, but before it starts playing. MacProcess can optionally invoke the following callbacks to alter some of the IDS data in the stream's session object: pfnAllocRtn and pfnUpdateRtn**.**

MOD will invoke an optionally defined MacReport with IAD billing information immediately after updating IAD.

MOD will invoke MacTerminate at shutdown.

Sample code

The **mod** project in the ITV tree contains a sample plug-in under the **service** project called **MODAuthorization**. This sample plug-in exercises each of the interfaces described in this document.

MOD Registry values for Plug-In Dll

The following registry values are reported via ManUtil.

NAME	TYPE	DESCRIPTION	DEFAULT
AuthorizationDllPath	REG_EXPAND_SZ	Location of the DLL	%ITVROOT%\Exe
AuthorizationDllName	REG_SZ	Name of the DLL	
AuthorizationDllTimeout	REG_DWORD	Number of milliseconds the MOD service will wait for a response from the DLL before granting or denying authorization	10000 milliseconds
Authorize On Error	REG_DWORD	Action to take when an authorization timeout occurs: 1 – authorize; 0 – don't authorize	1
AuthorizationDllThreadPoolMin	REG_DWORD	Minimum number of threads to be maintained in the pool	10
AuthorizationDllThreadPoolMax	REG_DWORD	Maximum number of threads allowed in the pool at any given time	20
AuthorizationDllThreadPoolGrowBy	REG_DWORD	Number of threads to grow by when the pool is exhausted	5
AuthorizationDllThreadPoolIdleTime	REG_DWORD	The number of milliseconds a thread can remain idle before it is released, provided that in doing so, the number of threads will not drop below the minimum	120000 milliseconds
AuthorizationDllThreadPoolReleaseTime	REG_DWORD	Number of milliseconds to wait for each outstanding thread during shutdown.	5000 milliseconds
AuthorizationDllThreadPoolStartTime	REG_DWORD	Number of milliseconds to wait for a new thread to start	60000 milliseconds

NOTE: By not specifying a value for AuthorizationDllName, you are instructing MOD to authorize all requests.

MacInitialize

Prototype

ITVSTATUS *MacInitialize* (MACHANDLES *pHandles,
MACCALLBACKS *pCallbacks)

Description

This function performs any initialization the plug-in wishes to make.

Parameters

MACHANDLES	IN	Address of structure used to pass 1.) a management package handle to the DLL. 2.) the instance ID of the invoking MOD 3.) a count of IDS data available to MacProcess 4.) an array of IDS data available to MacProcess 5.) a count of IAD data available to MacReport 6.) an array of IAD data available to MacReport
MACCALLBACKS	IN	Address of structure used to pass 1.) an unload callback so that the DLL can signal MOD to unload itself. 2.) an allocate callback so that the DLL can create a TLV. MOD will free the TLV. 3.) an update callback so that the DLL can update a subset of IDS data in a stream's session object.

Returns

This function returns a status code from the ITV platform error set.

The DLL should set the bRequested flag in the IDS and IAD arrays for desired data. All IDS array entries flagged as 'requested' will be supplied on each MacProcess invocation. All IAD array entries flagged as 'requested' will be supplied on each MacReport invocation.

If the DLL invokes the unload callback, MOD will behave as though the DLL had never been loaded; it will automatically authorize all subsequent requests. A warning is written to the NT Event Log when this happens and an alarm will be generated to let the operator know that authorization checking has stopped.

MacProcess

Prototype

ITVSTATUS *MacProcess* (BYTE *pRequest,
BOOL *pbResponse)

Description

This function determines whether or not a client is authorized to stream a movie.
It is given data about the request on which it is expected to perform the check.
It must accept or reject the request by setting the response value to TRUE or FALSE.

Parameters

pRequest	IN	TLV ("Type-Length-Value") byte stream containing: MacAddress, BillingId, SmartCardId, HomeId, StreamId and IDS data requested by MacInitialize.
pbResponse	IN OUT	The address of variable in which the result is placed.

Returns

This function returns a status code from the ITV platform error set. If the return value is ITV_SUCCESS and *__pbResponse__ is TRUE, MOD will accept the request. If the return value is not ITV_SUCCESS or if *__pbResponse__ is FALSE, MOD will reject the request.

The MOD service will track the minimum, mean and maximum amount of time it takes to process authorization checks and report them via ManUtil.

While in this function, the DLL can override certain IDS data for the current session by constructing a TLV of the data to be updated and passing it back to MOD. The TLV should be allocated using the pfnAllocRtn callback. Invoking the pfnUpdateRtn callback instructs MOD to update the session and release the TLV memory.

NOTE: MacProcess is a synchronous call. If MacProcess fails to return within a configurable amount of time, MOD can be configured to temporarily grant authorization. When MacProcess returns, MOD will terminate the stream if the response indicates an authorization failure. The release code of a terminated stream due to an authorization failure after a timeout is the same as that of a rejected stream due to an authorization failure.

WARNING: This function must be thread-safe; MOD can use a configurable number of threads to invoke it, i.e. 10 concurrent requests are received, 10 concurrent MacProcess invocations can be made.

MacReport

Prototype

ITVSTATUS *MacReport* (BYTE *pRequest,
BOOL *pbResponse)

Description

This function supplies IAD billing data for a terminated stream.
It is given data about the request on which it is expected to perform the check.
It should acknowledge the request by setting the response value to TRUE or FALSE.

Parameters

pRequest	IN	TLV ("Type-Length-Value") byte stream containing: StreamId and IAD data requested by MacInitialize.
pbResponse	IN OUT	The address of variable in which the result is placed.

Returns

This function returns a status code from the ITV platform error set.

The MOD service will track the minimum, mean and maximum amount of time it takes to process 'reports' and display them via ManUtil.

NOTE: MacReport is a synchronous call. MOD service uses the same thread to update IAD as it does to invoke MacReport. Delays in processing a MacReport request will impact MOD's updating of IAD.

See mod.h and dsmccdef.h for session release codes.
Release code examples:

1. 0x8004 - cm_UNSession_RsnItvUser; Client initiated suspend due to user request
2. 0xbffffffd - MOD_TERM_DURATION_TIMEOUT; MOD service release due to expiration of rental time

Request Details for MacProcess

DataType	Element	Description
BYTE	Request Version	MOD Authorization Check Version; value is 2
WORD	Descriptor Count	The number of descriptors to follow; value is 5+n where n is the number of requested IDS data fields
WORD	MacAddress Tag	MacAddress Descriptor Tag; value is 1
WORD	MacAddress Length	MacAddress Descriptor Length
BYTE[]	MacAddress Value	MacAddress
WORD	Billing ID Tag	Billing ID Descriptor Tag; value is 2
WORD	Billing ID Length	Billing ID Descriptor Length; value is 4
DWORD	Billing ID	Billing ID for SA systems
WORD	Home ID Tag	Home ID Descriptor Tag; value is 3
WORD	Home ID Length	Home ID Descriptor Length; value is 4
DWORD	Home ID	Home ID
WORD	SmartCard ID Tag	SmartCard ID Descriptor Tag; value is 4
WORD	SmartCard ID Length	SmartCard ID Descriptor Length; value is 4
DWORD	SmartCard ID	SmartCard ID
WORD	Stream ID Tag	Stream ID Descriptor Tag; value is 5
WORD	Stream ID Length	Stream ID Descriptor Length; value is 12
STREAMID	Stream ID	Stream ID

All Tags and Lengths for the subsequent IDS data have a data type of WORD.

DataType	Description	Tag Value	Updatable
DWORD	Asset’s length in minutes	21	N
DOUBLE	Asset’s price	22	N
DOUBLE	Asset’s analog copy charge	23	N
DWORD	Asset’s analog copy allowed flags	24	N
BYTE	Asset’s VCR commands allowed flag	25	Y
DOUBLE	Asset’s VCR charge	26	N
DOUBLE	Asset’s computed price	27	Y
DWORD	Asset’s rental time in minutes	28	Y
DWORD	Asset’s preview time in minutes	29	Y
WCHAR[]	Asset’s content provider	30	N
WCHAR[]	Asset’s genre(s) (comma-delimited)	31	N
WCHAR[]	Asset’s short title	32	N
DWORD	Asset’s rental type	33	N
DWORD	Asset’s viewing time in minutes	34	Y
DWORD	Asset’s viewing type	35	N
BYTE	Asset’s scrambled flag	36	N
DWORD	Asset’s scrambled time	37	N
DWORD	Asset’s scrambled type	38	N
WCHAR[]	Asset’s event ID	39	N
DWORD	Asset’s deactivate time	40	N
BYTE	Asset’s ‘can be suspended’ flag	41	Y
WCHAR[]	Asset’s provider	42	N
WCHAR[]	Asset’s rating(s) (comma-delimited)	43	N
WCHAR[]	Asset’s title	44	N
WCHAR[]	Asset’s brief title	45	N
WCHAR[]	Asset’s provider ID	46	N
WCHAR[]	Asset’s provider asset ID	47	N
DOUBLE	Folder’s pricing options for this asset	48	N
DWORD	Folder’s pricing value for this asset	49	N
DWORD	Folder’s suspend list options for this asset	50	N
DOUBLE	Folder’s suspend list lifetime for this asset	51	N
DWORD	Folder’s package ID for this asset	52	N
WCHAR[]	Folder’s billing tag for this asset	53	N

Request Details for MacReport

DataType	Element	Description
BYTE	Request Version	MOD Authorization Check Version; value is 1
WORD	Descriptor Count	The number of descriptors to follow; value is 1+n where n is the number of requested IAD data fields
WORD	Stream ID Tag	Stream ID Descriptor Tag; value is 5
WORD	Stream ID Length	Stream ID Descriptor Length; value is 12
STREAMID	Stream ID	Stream ID

All Tags and Lengths for the subsequent IAD data have a data type of WORD.

DataType	**Description**	**Tag Value**
DWORD	Billing record type	101
WCHAR[]	Session’s MacAddress	102
DWORD	Session’s billing ID	103
DWORD	Asset’s ID	104
WCHAR[]	Asset’s title	105
WCHAR[]	Asset’s short title	106
BYTE	Session’s billable flag	107
WCHAR[]	Asset’s provider	108
DWORD	Asset’s analog copy allowed flags	109
WCHAR[]	Asset’s genre(s) (comma-delimited)	110
DWORD	Session’s fast-forward count	111
DWORD	Session’s rewind count	112
DWORD	Session’s pause count	113
DWORD	Session’s play time in milliseconds	114
DOUBLE	Asset’s computed price	115
DWORD	Session’s purchase time	116
DWORD	Session’s accumulated seconds within rental period	117
DWORD	Session’s Home ID	118
DWORD	Session’s SmartCard ID	119
DWORD	Session’s Purchase ID	120
WCHAR[]	Asset’s event ID	121
DWORD	Session’s package ID	122
WCHAR[]	Asset’s provider	123
WCHAR[]	Asset’s rating(s) (comma-delimited)	124
DOUBLE	Asset’s VCR charge	125
DOUBLE	Asset’s analog copy charge	126
DWORD	Session’s context ID	127
WCHAR[]	Asset’s provider ID	128
WCHAR[]	Asset’s provider asset ID	129
BYTE	Session’s new purchase flag	130
DWORD	Session’s release code	131

STREAMID Definition

DataType	**Element**	**Description**
Long	dwStreamIdNumber	Unique stream number for a CM.
Long	Inst	Inst of the CM.
Long	Type	Type of the CM.

*Keep in mind that as additional information about a request is added to the ITV System, new descriptors may be added. Therefore, your code should skip any descriptors it does not know and continue parsing the request.

MacUninitialize

Prototype

ITVSTATUS *MacUninitialize* ()

Description

This function performs any cleanup the plug-in wishes to take.

NOTE: After a request to unload the DLL has been made (either by the DLL or MOD), MOD will not make any more MacProcess calls and it will wait for all outstanding MacProcess calls to complete before invoking MacUninitialize.

Parameters

None.

Returns

This function returns a status code from the ITV platform error set.

Data Structures

```
// From SeaInclude.h; A version structure for specifying the version of a product, exe, dll, lib, etc
typedef union _ITVVERSION {
   WORD      wVersion;    // The whole version, can compare.
   struct
   {
      BYTE    byteMinor;  // Minor version, 0-99
      BYTE    byteMajor;  // Major version, 0-255
   } VersionComponents;
} ITVVERSION, *P ITVVERSION;

// From mac_interfaces.h; These header file defines the interface between MOD and the DLLs that it loads

// Version Number for Structures
#define MAC_MAJOR_VERSION      2
#define MAC_MINOR_VERSION      0
#define MAC_VERSION_2_0        0x0200

typedef struct _MACINFO {
   BOOL   bRequested;      // Does the DLL want this data when MAC_PROCESS_RTN or
                           // MAC_REPORT_RTN is invoked?
   WCHAR  wszName[255];    // Name
   WORD   wTag;            // See request tags below
} MACINFO;

typedef struct _MACHANDLES {
   ITVVERSION  Version;          // Version of structure
   HANDLE      hMgmt;            // Management handle for DLL to provide mgmt interface
   WORD        wID;              // Instance Id of MOD
   WORD        wDsInfo;          // The number of DS values in ppDsInfo
   MACINFO     **ppDsInfo;       // DS data requested when invoking MAC_PROCESS_RTN
   WORD        wAdInfo;          // The number of AD values in ppDsInfo
   MACINFO     **ppAdInfo;       // AD data requested when invoking MAC_REPORT_RTN
} MACHANDLES;

typedef BOOL (__cdecl *MAC_UNLOAD_RTN) (void);
typedef BYTE *    (__cdecl *MAC_ALLOC_RTN)  (DWORD dwLength);
typedef ITVSTATUS (__cdecl *MAC_UPDATE_RTN) (STREAMID Sid,
                                             BYTE *pRequest,
                                             BOOL *pbResponse);
typedef struct _MACCALLBACKS {
   ITVVERSION      Version;       // Version of structure
   MAC_UNLOAD_RTN  pfnUnloadRtn;  // Causes DLL to be unloaded
   MAC_ALLOC_RTN   pfnAllocRtn;   // Used by DLL to allocate TLV used by MAC_UPDATE_RTN
   MAC_UPDATE_RTN  pfnUpdateRtn;  // Used by DLL to update Session data
} MACCALLBACKS;
```

```
#define MAC_TAG_MACADDRESS          1
#define MAC_TAG_BILLINGID           2
#define MAC_TAG_HOMEID              3
#define MAC_TAG_SMARTCARDID         4
#define MAC_TAG_STREAMID            5

 typedef ITVSTATUS (__cdecl *MAC_INIT_RTN) (MACHANDLES *pHandles,
                                            MACCALLBACKS *pCallbacks);
typedef ITVSTATUS (__cdecl *MAC_UNINIT_RTN)  (void);
typedef ITVSTATUS (__cdecl *MAC_PROCESS_RTN) (BYTE *pRequest,
                                              BOOL *pbResponse);

#define MAC_TAG_DS_ASSET_LENGTH                          21
#define MAC_TAG_DS_ASSET_PRICE                           22
#define MAC_TAG_DS_ASSET_ANALOG_COPY_CHARGE              23
#define MAC_TAG_DS_ASSET_ANALOG_COPY_ALLOWED             24
#define MAC_TAG_DS_ASSET_VCR_ALLOWED                     25
#define MAC_TAG_DS_ASSET_VCR_CHARGE                      26
#define MAC_TAG_DS_ASSET_COMPUTED_PRICE                  27
#define MAC_TAG_DS_ASSET_RENTAL_TIME                     28
#define MAC_TAG_DS_ASSET_PREVIEW_TIME                    29
#define MAC_TAG_DS_ASSET_CONTENT_PROVIDER                30
#define MAC_TAG_DS_ASSET_GENRE                           31
#define MAC_TAG_DS_ASSET_SHORT_TITLE                     32
#define MAC_TAG_DS_ASSET_RENTAL_TYPE                     33
#define MAC_TAG_DS_ASSET_VIEWING_TIME                    34
#define MAC_TAG_DS_ASSET_VIEWING_TYPE                    35
#define MAC_TAG_DS_ASSET_SCRAMBLED                       36
#define MAC_TAG_DS_ASSET_SCRAMBLED_TIME                  37
#define MAC_TAG_DS_ASSET_SCRAMBLED_TYPE                  38
#define MAC_TAG_DS_ASSET_EVENT_ID                        39
#define MAC_TAG_DS_ASSET_DEACTIVATE_TIME                 40
#define MAC_TAG_DS_ASSET_SUSPENDABLE                     41
#define MAC_TAG_DS_ASSET_PROVIDER                        42
#define MAC_TAG_DS_ASSET_RATING                           43
#define MAC_TAG_DS_ASSET_ASSET_TITLE                     44
#define MAC_TAG_DS_ASSET_ASSET_BRIEF_TITLE               45
#define MAC_TAG_DS_ASSET_PROVIDER_ID                     46
#define MAC_TAG_DS_ASSET_PROVIDER_ASSET_                 47

#define MAC_TAG_DS_FOLDER_PRICINGOPTIONS                 48
#define MAC_TAG_DS_FOLDER_PRICINGVALUE                   49
#define MAC_TAG_DS_FOLDER_SUSPENDLISTOPTIONS             50
#define MAC_TAG_DS_FOLDER_SUSPENDLISTLIFETIME            51
#define MAC_TAG_DS_FOLDER_PACKAGEID                      52
#define MAC_TAG_DS_FOLDER_BILLINGTAG                     53
```

```
#define MAC_TAG_AD_TYPE                             101
#define MAC_TAG_AD_MAC_ADDRESS                      102
#define MAC_TAG_AD_BILLING_ID                       103
#define MAC_TAG_AD_ASSET_ID                         104
#define MAC_TAG_AD_TITLE                            105
#define MAC_TAG_AD_SHORT_TITLE                      106
#define MAC_TAG_AD_BILLABLE                         107
#define MAC_TAG_AD_ASSET_PROVIDER                   108
#define MAC_TAG_AD_ANALOG_COPY_ALLOWED              109
#define MAC_TAG_AD_ASSET_GENRE                      110
#define MAC_TAG_AD_FAST_FORWARD_COUNT               111
#define MAC_TAG_AD_REWIND_COUNT                     112
#define MAC_TAG_AD_PAUSE_COUNT                      113
#define MAC_TAG_AD_PLAY_TIME                        114
#define MAC_TAG_AD_PRICE                            115
#define MAC_TAG_AD_PURCHASE_TIME                    116
#define MAC_TAG_AD_RENTAL_TIME                      117
#define MAC_TAG_AD_HOME_ID                          118
#define MAC_TAG_AD_SMART_CARD_ID                    119
#define MAC_TAG_AD_PURCHASE_ID                      120
#define MAC_TAG_AD_EVENT_ID                         121
#define MAC_TAG_AD_PACKAGE_ID                       122
#define MAC_TAG_AD_PROVIDER                         123
#define MAC_TAG_AD_RATING                           124
#define MAC_TAG_AD_VCR_CHARGE                       125
#define MAC_TAG_AD_COPY_CHARGE                      126
#define MAC_TAG_AD_CONTEXT_ID                       127
#define MAC_TAG_AD_PROVIDER_ID                      128
#define MAC_TAG_AD_PROVIDER_ASSET_ID                129
#define MAC_TAG_AD_NEW_PURCHASE                     130
#define MAC_TAG_AD_RELEASE_CODE                     131
```