# TiVo IT Web Services Guide

## version 9


*© 2012 • TiVo® Proprietary and Confidential*

© 2012 TiVo Inc. Reproduction in whole or in part without written permission is prohibited. All rights reserved. TiVo, the TiVo logo, TiVo Central, WishList, Season Pass, TiVoToGo, the Advance logo, the Replay logo, the Thumbs Up logo, the Thumbs Down logo, Overtime Scheduler, Overlap Protection, Ipreview, and the sounds used by the TiVo Service are trademarks or registered trademarks of TiVo Inc. or its subsidiaries, 2160 Gold Street, P.O. Box 2160, Alviso, CA 95002-2160.

CableCARD™ is a trademark of Cable Television Laboratories, Inc.

Manufactured under license from Dolby Laboratories. “Dolby” and the Double-D symbol are trademarks of Dolby Laboratories.

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC in the United States and other countries.

All other trademarks are the properties of their respective owners.

U.S. Patent Nos. 6,215,526; 6,233,389; 6,310,886; 6,327,418; 6,385,739; 6,487,646; 6,490,722; 6,535,253; 6,642,939; 6,643,798; 6,728,713; 6,757,837; 6,757,906; 6,792,195; 6,847,778; 6,850,691; 6,868,225; 6,965,730; 7,055,031; 7,158,531; 7,200,321; 7,216,235; 7,228,556; 7,321,716; 7,409,546; 7,484,234; 7,493,015; 7,529,465; 7,543,325; 7,558,472; 7,590,240; D434,043; D435,561; D445,801; D517,059. 5,315,448; 6,381,747; 6,516,132.

TiVo remote control: U.S. Pat. Nos. D424,061; D424,577; D431,552; D433,403; D463,788.

Other patents pending.

# Table of Contents

# 8 Online Service

## 9 XML Objects

# Version History

1

## What's new in this release

Version 9 of the TiVo web services adds the following:

- Ability to call explicit version of a web service operation. For more information, see *Web Services Versioning* *on page 23*.
- Several new elements added to the *configInfo Field Group* on page 94.
- New `tier` element added to *deviceFieldGroup* on page 111.
- New `manufacturingInfo` element added to *deviceInfoSearchResponse* on page 113.
- New XML element added in *manufacturingInfo* on page 122.

### Version 9

The book version number is now synchronized with the web services version. Thus, this book is version 9 and documents version 9 of the web services. The document was previously referred to as version 1.12.

The following table lists all the changes in the book since the previous release.

| Change | Description |
|---|---|
| Chapter 2, Overview | Reorganized slightly and added "Overview of the Web Services" (section 2.2). |
| Chapter 3, Using the Services | Added section 3.1, *Setup*.<br>Added section 3.2.3, *Web Services Versioning.* |
| Chapter 5, Account Service | Section 5.3, *customerInfoStore Operation*: added clarification to operation description. |
| Chapter 5, Account Service<br>Chapter 6, Commerce Service<br>Chapter 7, Device Service<br>Chapter 8, Online Service | Moved operation inputs and outputs into their own columns in tables. |
| Chapter 7, Device Service | Section 7.3, *ConfigInfoSearch Operation*: added URL request and response for version 9 web services.<br>Section 7.4.2 *Output Element* (*deviceInfoSearch Operation*): added Sample XML Response from Version 9 Web Service. |

| Change | Description |
|---|---|
| Chapter 9, XML Objects | Corrected section numbering.<br>In *configInfo Field Group* on page 94: added new elements returned by version 9 web services.<br>In *deviceFieldGroup* on page 111: added new tier element.<br>In *deviceInfoSearchResponse* on page 113: added new manufacturingInfo element<br>New section: *manufacturingInfo* on page 122. |
| Error codes chapter | Added:<br>50027 Update could not be applied because of device tier.<br>60200 Operation is not valid because of device tier. |

# Previous versions

## Version 1.11b

| Change | Description |
|---|---|
| Added paragraph about services disclaimer. | Added to section 2.1. |
| Change tws and twsqa references in URLS. | URLS on TomCat are otws and otwsqa. |
| Remove beta environment references | |
| Chapter 5 | Added resetPassword. |
| Error codes chapter | Added:<br>23001 Error while sending the email<br>23002 Customer not found<br>23100 There are missing parameters or bad values in the request. Missing or invalid Parameters:<br>3004 - Customer not found for the given Partner Customer ID |
| | Changed the sample xml input for the customerInfoStore |
| | Changed the sample xml input for the serviceActivate |
| | Changed the sample xml input for the deviceInfoStore |
| | Changed the second sample XML input for the callInfoSearch |
| | Changed the two example XML inputs for the dialCodeUpdate |
| In Section 2.3 | Modified the 5 "TiVo IT Services Environment" diagrams to remove the blue line that goes from the TiVo Web Services QA box to the Staging Broadcast Center box. |
| In Section 5.3 | Changed the "Note:" |
| In Section 6.2 (serviceActivate Operation) | Removed the "Note: " statement. |

## Version 1.11

| Change | Description |
|---|---|
| Section 2 | Overview material, with conceptual illustrations, is entirely new. |
| Sections 5 through 8 | Added input paramter and output element notations preceding all example code. |
| Section 9 | Added contextual usage information and extensive cross-references to the XML Object descriptions. |
| Section 9 | Added Read/Write annotations to the XML object tables. |

| Change | Description |
|---|---|
| Section 9 | Removed obsolete XML object references:<br>float<br>id:body<br>id:Station<br>id:Subscription<br>idSearch<br>incident<br>incident Field Group<br>localTimeOffset<br>objectId28<br>transactionResponse Field Group<br>transactionResponse |

## Version 1.10

| Change | Description |
|---|---|
| headendID | Changed headendID to msoServiceID |
| privacyStatus | Changed optOut description. |
| Devices | Chapter 4, all new material. |
| serviceActivate operation | Updated sect 6.1 to include new sample input and sample URL. |
| deviceInfoStore | Updated sect 7.4 to include new sample input and sample URL. |
| privacyLevel options | Updated 9.63 to include new privacy level settings |
| Index | Created index. |

## Version 1.9

| Change | Description |
|---|---|
| Section 4.1 | The TiVo Service Number (TSN) must contain exactly 15 hexadecimal characters [a-fA-F0-9] |
| Section 7.9 | New/revised code examples for serviceMessageStore operation. |
| Section 7.11 | New/revised code examples for serviceMessageSearch operation. |
| Section 9.64 | Added partnerContentId and partnerCollectionId fields to recording object. |
| Section 9.90 | Updated field list for serviceMessageFieldGroup |
| Section 9.91 | Added serviceMessageList. |
| Section 9.92 | Deleted serviceMessageSearchResponseList. |
| Section 9.92 | Added serviceMessageSearch Field Group. |
| Section 9.93 | Updated field list for serviceMessageSearchRepsonse |

## Version 1.8

| Change | Description |
|---|---|
| Section 8.2 | Added contentSearch operation. |
| Section 8.4 | recordingSearch operation<br>Changed bodyID to tivoSerialNumber.<br>Added partnerStationId, partnerCollectionId, partnerContentId. |
| Section 8.5 | scheduleRecording operation<br>Added partnerContentId, partnerStationId.<br>&partnerStationId |
| Section 9 | Changed object28 to objectID28. |
| Section 9 | Changed privacy status options. |

| Change | Description |
|---|---|
| Section 10 | Added error codes. |
| Section 9 | Added Message API. |
| Section 9 | Added dialCodeUpdate API. |
| Sections 3 through 8 | Changed XML samples to alphabetize elements. |
| Section 7.2 | callInfoSearchOperation<br>Changed XML response. |
| Section 7.3 | configInfoSearchOperation<br>Changed XML response. |

## Version 1.7

| Change | Description |
|---|---|
| Section 8.2 | Added contentSearch operation. |
| Section 8.4 | recordingSearch operation<br>Changed bodyID to tivoSerialNumber.<br>Added partnerStationId, partnerCollectionId, partnerContentId. |
| Section 8.5 | scheduleRecording operation<br>Added partnerContentId, partnerStationId.<br>&partnerStationId |
| Section 9 | Changed object28 to objectID28. |
| Section 9 | Changed privacy status options. |
| Section 10 | Added error codes. |
| Section 9 | Added Message API. |
| Section 9 | Added dialCodeUpdate API. |
| Sections 3 through 8 | Changed XML samples to alphabetize elements. |
| Section 7.2 | callInfoSearchOperation<br>Changed XML response. |
| Section 7.3 | configInfoSearchOperation<br>Changed XML response. |
| Section 7.4 | deviceInfoSearchOperation<br>Changed XML response. |
| Section 9.9 | configInfoFieldGroup<br>Changes to field descriptions. |
| Section 9.38 | drive<br>Changes to field descriptions. |
| Section 9.64 | networkInformation Field Group<br>Changes to field descriptions. |
| Section 10, error 33111 | Changed the description. |

## Version 1.6

| Change | Description |
|---|---|
| Section 4.2 | Device Service States<br>• Changed 'Canceled' to 'Canceled or Closed' in table of service states. |

| Change | Description |
|---|---|
| Section 5.1 | customerInfoSearch Operation<br>• Updated customerInfoSearch code examples to include optStatus element. |
| Section 5.2 | customerInfoStore Operation<br>• Updated customerInfoStore code examples to include optStatus element. |
| Section 6.1 | serviceActivate Operation<br>• Updated serviceActivate code examples to include siteId element. |
| Section 7.3 | deviceInfoSearch Operation<br>• Updated deviceInfoSearch code examples to include siteId element. |
| Section 7.4 | deviceInfoStore Operation<br>• Updated deviceInfoStore code examples to include siteId element. |
| Section 8 | Online Service<br>• Deleted idSearch operation.<br>• Added recordingSearch operation. |

## Version 1.5

| Change | Description |
|---|---|
| Section 3.0 | Replaced "Partners' Customer ID" with "PartnerCustomerID" and "customer email" with "CustomerEmail"<br>Added text: In most cases, the default setting is optNeutral. |
| Section 3.1 | Added emphasis: When POSTing, sibling elements in the XML document must be listed in alphabetical order. |
| Section 4.2 | Revised text: Each TiVo device is assigned a service state that changes during the device life cycle. For example, when a device is manufactured, it is initialized to a service state of New. When the TiVo service is activated for a limited service term (monthly or yearly, for instance), the service state is set to Good. When the TiVo service is activated for lifetime service, the service state is set to Product Lifetime.Revised text: Note: When the Commerce Service is initiated, the new service state is available to the device within 15 minutes. The device will receive its updated service state after it makes a service connection. |
| Section 4.3 | Updated info in the Service State Transitions table. |
| Section 6.1 | Revised description: Activates a device and makes the TiVo service available to it; creates a customer if this is the first time the customer has activated a device. |
| Section 6.2 | Added info (serviceCancel): A customer record is never deleted, even if there are no active devices for that customer. |
| Section 6.3 | Added info (serviceDelete): A customer record is never deleted, even if there are no active devices for that customer. |

## Version 1.4

| Change | Description |
|---|---|
| Section 1 | New Version History topic describes changes in document versions. |
| Section 3.<br>Invoking an Operation | Included a statement and code example to indicate that when POSTing, sibling elements in the XML document must be listed in alphabetical order. |
| Section 6.1<br>`serviceActivate` Operation | In the sample XML input, removed the following element:<br>`<activationDate>5-1-2010</activationDate>` |
| | In the sample URL, removed:<br>&contract.0.partnerOrderNumber=12312<br>&contract.0.activationDate=5-1-2010 |
| Section 7Device Service | In the Operations table, the input element for the `deviceInfoStore` operation should just be `device`. Not `device, Device info`. |

| Change | Description |
|---|---|
| Section 7.2<br>`configInfoSearch` Operation | In the first sample URL, `tivoSerialNumber` was duplicated. |
| Section 7.4<br>`deviceInfoStore` Operation | Only the following fields can be updated:<br>• `NAME` (device name)<br>• `CDS`<br>• `ESATA`<br>• `HME`<br>• `HTTPAccess`<br>• `MRV`<br>• `Music Photo`<br>• `TCO`<br>• `TTCB`<br>• `TTG` |
| Section 8.1<br>How Remote Scheduling Works | Removed the statement, "A scheduling request delivered to the device..." |
| | Changed the operation name from `captureRequest` to `schedulingRecordingStore`. |
| Section 9<br>XML Objects | Added a column to indicate whether a field is required. |
| Section 9.11<br>`contractFieldGroup` | Removed the following fields:<br>• `salesChannel`<br>• `activationDate`<br>• `partnerOrderNumber`<br>• `material`<br>• `referrerEmail` |
| Section 9.57<br>`scheduleRecordingFields` | Updated the following field descriptions. |
| `partnerCustomerId` | Description (applies to all `partnerCustomerId` fields): Up to 20 characters. Identifies a customer within the TiVo system. Should be unique for each customer. |
| `tivoSerialNumber` | Description (applies to all `tivoSerialNumber` fields): 15-digit alphanumeric string. Unique for each device. |
| `channelNumber` | Type: string. Same channel number as in the line-up. |
| `email` | Type: string. Email address to send recording notification. |
| `firstEmailFlag` | Type: string. For broadband connected devices, the first email is N/A. Default value is `false`. |
| Section 9.71<br>`showStatus` | Changed the description to "See `showType`." |
| Section 9.72<br>`showType` | Added information to the description. |
| Section 10<br>Error Codes | New code: 31200<br>Message: Exception: *`Error message`* |

| Change | Description |
|---|---|
| | New code: 32200<br>Message: Exception: *Error message* |
| | New code: 33017<br>Message: This TSN has active contract: *contract number* |
| | New code: 50301<br>Message: Receiver name not unique for customer |

# Overview

2

## 2.1 About This Document

TiVo implements web services that partners can use to manage customer accounts, manage settings on customers' devices, and to schedule recordings. This document outlines the TiVo partner integration testing process, which is designed to ensure that partner applications invoke TiVo services properly. This document also describes how to use the services, explains how the services work, and provides a reference to the operations and data structures for each service.

### 2.1.1 Code Examples

Code examples (code listings and sample URLs) included throughout this document include long lines that wrap to two or more display lines. In many cases, line breaks or spaces have been inserted to make the wrapped code easier to read. Therefore you cannot always copy and paste code examples and expect them work without additional editing. For example, you may have to delete spaces in an example URL to make it work as expected.

### 2.1.2 Required and Optional Parameters

This document lists all available parameters for each described service operation. Parameters may be required or optional, partly depending on specific partner requirements and implementations. Separate documentation will be provided to partners to specify which parameters are required and which are optional for their implementations.

## 2.2 Overview of the Web Services

The TiVo web services are:

- **Account service:** use to work with customer account data.
- **Commerce service:** use to activate service on a device, cancel service on a device, and reset a device.
- **Device service:** use to retrieve a TiVo box's configuration and call history information. This service provides several operations useful to troubleshoot customer issues.
- **Online service:** use to request a recording of specific broadcast content on a specific device, search for content, and search on recordings.

### 2.2.1 Available Services and Operations

The services contained in this document are a catalog of available APIs that a TiVo partner can potentially use to interface with TiVo IT systems. All services are not available by default. Rather,

TiVo defines the actual services and operations available during the partner implementation and lists them in the IT Release Notes it provides to the partner.

In general, the following services and operations are provided by default to enable provisioning and diagnostics capabilities:

Account service:

- customerInfoSearch
- customerInfoStore

Commerce Service

- serviceActivate
- serviceCancel
- serviceCancelByCustomerId
- serviceReset

Device service:

- callInfoSearch
- configInfoSearch
- deviceInfoSearch
- deviceInfoStore

### 2.2.2 Security Measures

Partner web services make use of digital certificates and keys (public and private) to ensure transaction security. All information passed between partners and TiVo is sent using a secure protocol (HTTPS) to ensure that the data is encrypted when sent across public networks.Two-way SSL is used for communication between partners and TiVo. A partner must present a TiVo-generated and TiVo-signed SSL client certificate with each SSL connection request to the web services.

**Note:** Partners may be able to start out using a “demo account” certificate before receiving their actual, partner-specific digital certificates from TiVo. Partners can contact their TiVo account manager for more information.

### 2.2.3 Digital Certificates

Partners must install digital certificates on the application servers their applications reference when invoking TiVo web services.

Partners must contact their TiVo account managers to request the following two certificates:

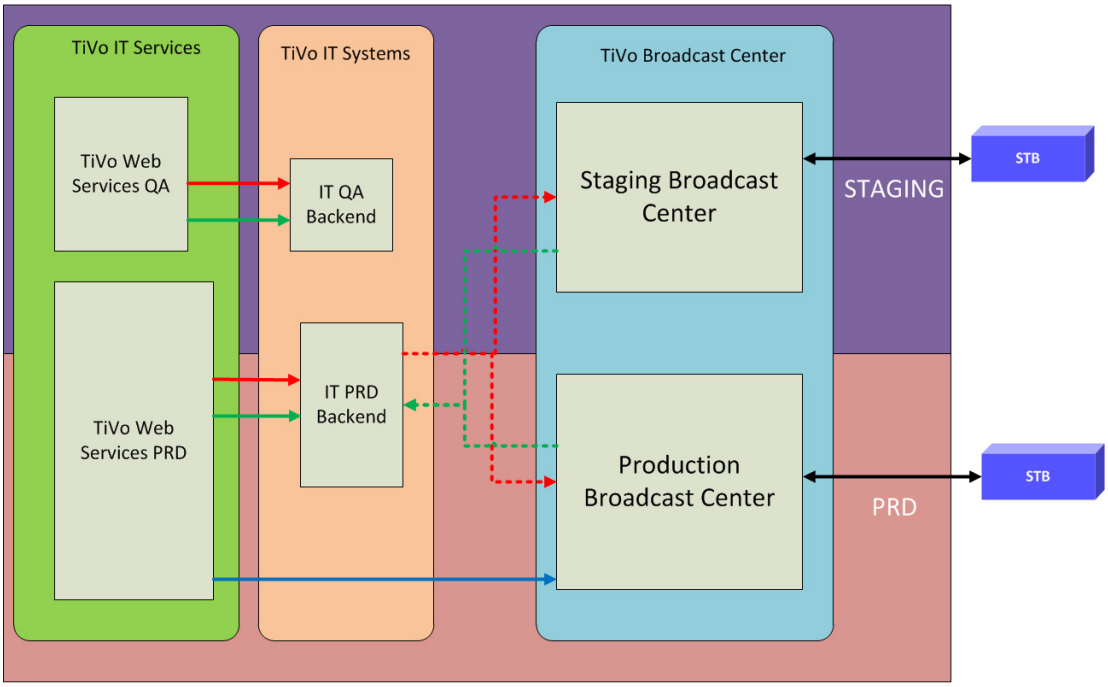
- One for interface testing with the TiVo testing environment.
- One for beta testing and production use.

For more information on installing certificates, see *Testing the partner digital certificate on page 19*.

## 2.3 Partner Integration Testing Process

The TiVo partner integration testing process ensures that partner applications invoke TiVo services properly. The testing process uses the TiVo IT Services environment:

**TiVo IT Services Environment**




**Legend:**
Data Flow for Provisioning Services (serviceActivate, seviceCancel, serviceReset, customerInfoStore, deviceInfoStore)
Data Flow for Support Services (customerInfoSearch, deviceInfoSearch, ...)
Data Flow for Content Discovery Services (recordingSearch, contentSearch)
Solid lines: real time updates
Dashed lines: scheduled updates

There are three stages to the testing process:

1. Testing and developing in the IT QA environment
2. Migrating, testing, and developing in the IT PRD environment
3. Going live in the PRD environment for partner launch

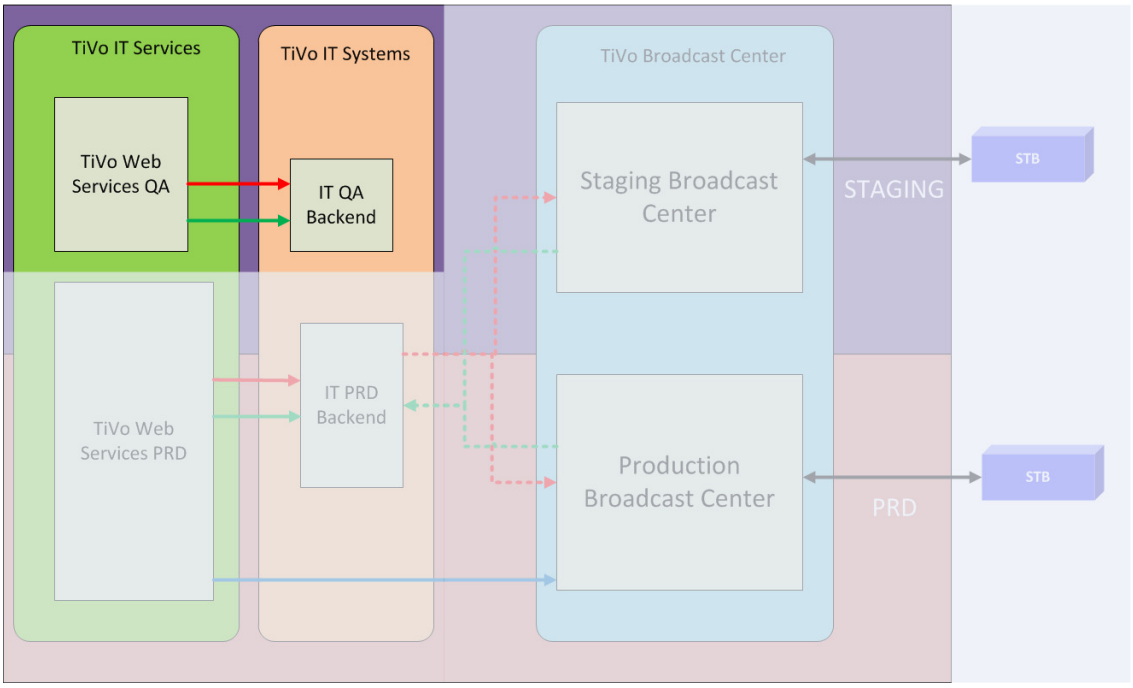
The Staging Broadcast Center is used prior to production launch to test developed services with TSNs that are associated with particular set-top boxes (STBs). The Production Broadcast Center is used when a partner launches. Whether a box dials into the Staging or Production Broadcast Centers is determined by a dial code. Partners can specify a dial code via an API call (`dialCodeUpdate`) or TiVo can specify it using internal tools.

### 2.3.1 Testing and Developing in the IT QA Environment

The partner testing integration process starts with testing and development in the TiVo web services QA environment. This is a non-production sandbox environment that allows the partner to experience both the certificate process and three essential TiVo web services.

Testing in the IT QA environment involves two stages: testing and development using a demo certificate that allows limited access to the services; and testing and development using a partner certificate. TiVo issues both kinds of certificates to the partner.

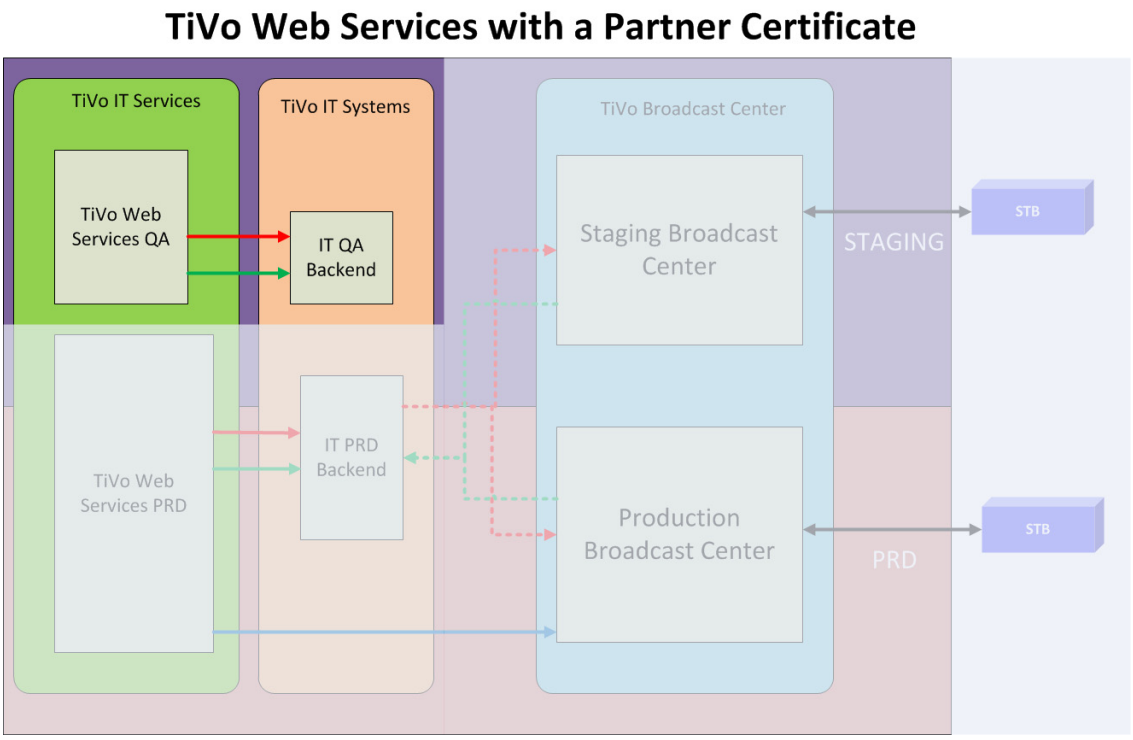
**TiVo Web Services with a Demo Certificate**




The first part of the testing integration process in the IT QA environment comprises the following steps:

1. The partner requests a digital certificate for access to the interface testing environment from their TiVo account manager.
2. TiVo issues a demo certificate and provides the partner with the URL of the TiVo interface testing environment which gives the partner access to three web services in the test environment: Account Service, Commerce Service, and Device Service.
3. The demo certificate is installed on the partner's test application servers.
4. The partner develops application software that references the digital certificate and interfaces with the TiVo web account, commerce, and device services.

Once these steps are complete, testing and developing in the IT QA environment advances to the next stage:

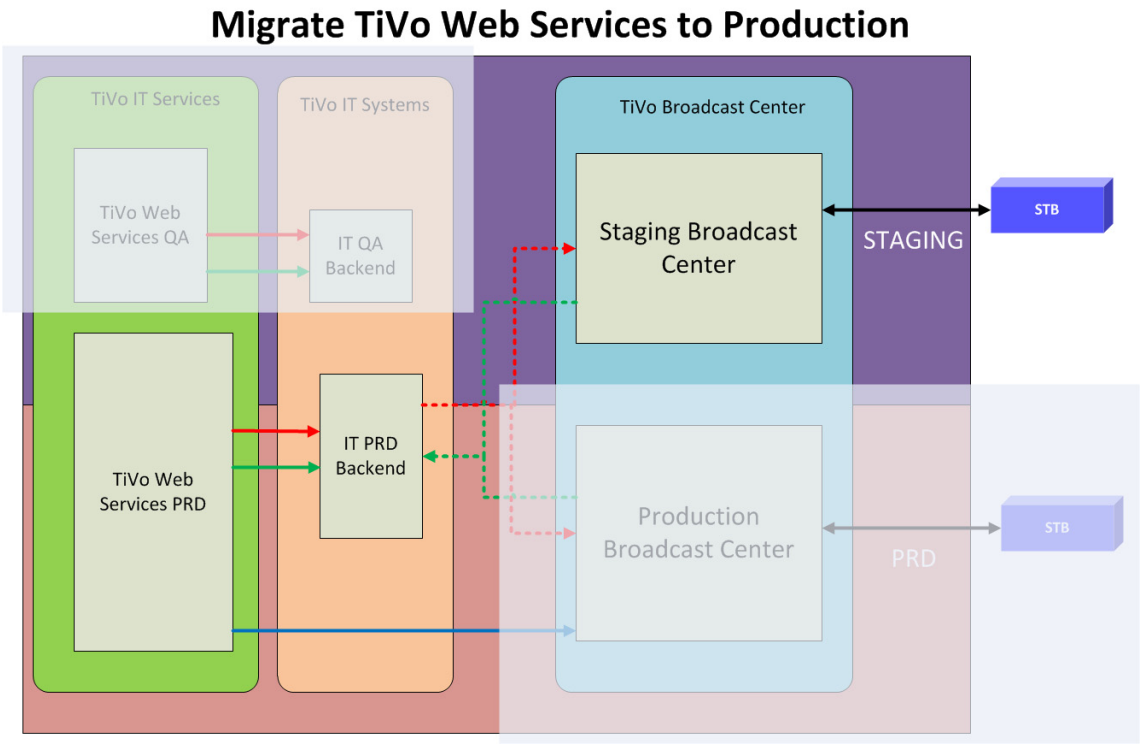



The second part of the testing integration process in the IT QA environment comprises the following steps:

1. TiVo issues the partner a partner certificate to replace the demo certificate.
2. The partner replaces the demo certificate on the partner's test application servers with the newly issued partner certificate.
3. TiVo develops any needed partner-specific web services and configures partner settings in the IT QA Backend.
4. The partner tests the TiVo web services using the partner certificate with the developed software.

## 2.3.2 Migrating, Testing, and Developing in the IT PRD Environment

Once initial testing and development in the IT QA environment is completed, testing can be migrated to the IT Production web services. TiVo recommends that partners perform end-to-end testing to devices connected to the Staging Broadcast Center prior to full production launch.




During this phase, TiVo migrates partner-specific web services from the IT QA environment to the IT web services production (PRD) environment. TiVo also makes partner-specific settings in the IT PRD backend, and in both the staging broadcast and production broadcast centers.

The staging broadcast center is used for field testing. The production broadcast center is readied for eventual partner launch.

Once TiVo has set up the web services for production testing, the partner uses the partner certificate and developed software in the production environment to interface with the TiVo web services prior to actual production launch.

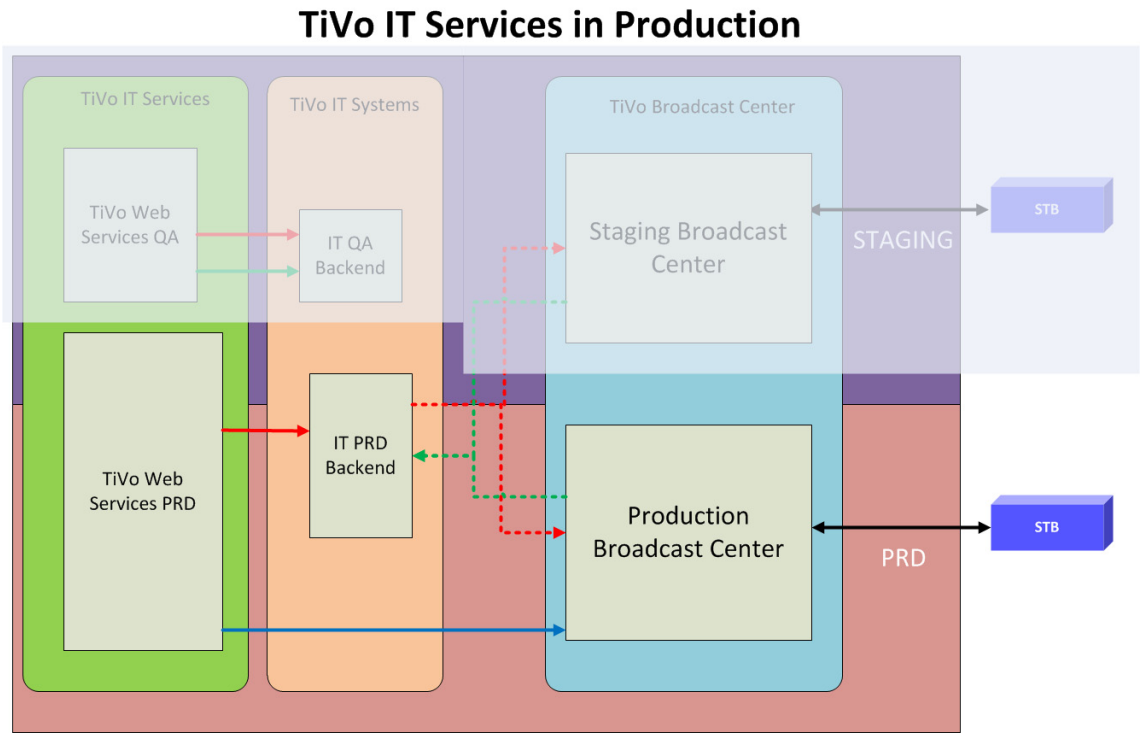
Testing at this stage is conducted against a list of TSNs provided by TiVo to the partner.

## 2.3.3 Going Live

When the partner has successfully tested their web services, developed and tested their applications, and tested their TSNs, the partner is ready to go live.

The partner should contact their TiVo account manager so that TiVo is aware of the partner's launch, and has partner contact information needed in the event of a post-launch escalation.

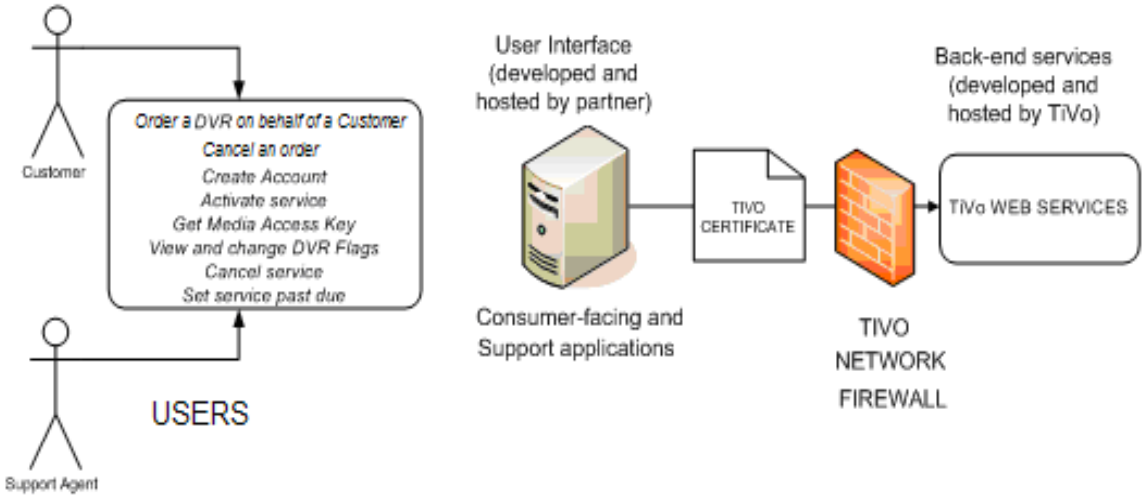
Finally, the partner then deploys the application to the partner's own production environment, configures the application to use the production certificate, and points to the production URL for the TiVo web services:



# Using the Services

3

The following diagram shows how a partner uses the TiVo IT web services in applications built for customers and support agents:




Before integrating applications with the TiVo web services, a partner works with a TiVo account manager to define the following business rules:

1. A unique identifier that represents a single customer: `PartnerCustomerID` or `CustomerEmail`.
2. Default setting for the customer account `optStatus` field, which indicates the degree to which customer viewing data may be collected and associated with a device. In most cases, the default setting is `optNeutral`.
3. Default settings for device flags including:
   - `downloadsEnabled`, which indicates whether to allow TiVoCast and similar downloaded content.
   - TiVo To Go (TTG), which enables customers to transfer programs from a device to a PC.
   - TiVo To Come Back (TTCB), which enables customers to transfer programs from a PC to a device.
   - Multi-room viewing (MRV), which enables customers to transfer programs from one device to another in the same account and on the same local area network.

**Note:** For a full list of device flags, see *deviceInfo* on page 112.

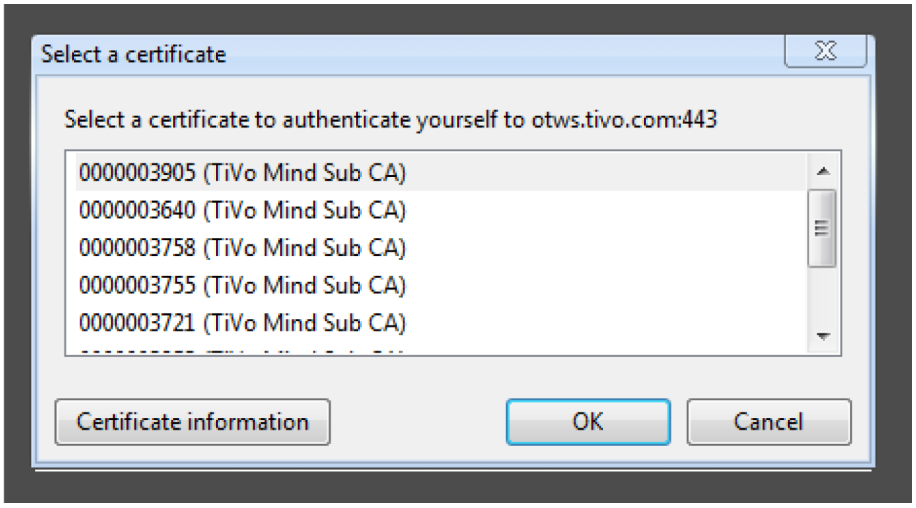
## 3.1 Testing the partner digital certificate

You must install a digital (public key) certificate to authenticate requests to TiVo web services and accept secure responses from them.

### 3.1.1 Procedure (Windows)

Follow these steps to initially setup your web service connection.

1. Obtain the certificate file from your TiVo account manager or project manager..
2. Double-click the certificate.
3. Follow the instructions from the install wizard.
4. In a web browser load this URL:
   `https://otwsqa.tivo.com/deviceInfoSearch`
   `?type=deviceInfoSearch`
   `&tivoSerialNumber=`*`tivoSerialNumber`*
   where *tivoSerialNumber* is a valid TSN. You can test by changing the TiVo serial number parameter to a test TSN.
5. The browser will prompt you to select a certificate, as illustrated in the figure. If you have installed other certificates, you must keep track of which certificate to use when using TiVo web services.




   The web browser will display the web service XML response.

Test the installation of the certificate by directly hitting the appropriate web service URL and supplying valid parameter data to receive a response.

### 3.1.2 Procedure (command line)

On systems that support it (such as Linux or Mac OSX), you can use the cURL command to make TiVo web service requests. This command is typically useful during development and testing.

For example:

```
curl https://otwsqa.tivo.com/deviceInfoSearch
?type=deviceInfoSearch
&tivoSerialNumber=tivoSerialNumber
```

# 3.2 Invoking a TiVo Partner Web Service Operation

You can invoke a TiVo Partner Web Service operation using POST or GET. An application can either POST an XML document via HTTPS, or submit an HTTPS GET request to specify parameters and values in the URL in lieu of an XML document.

There are multiple versions of the TiVO web services available. In either case, you may request a specific version of web service if applicable.

## 3.2.1 Using POST

POST requests are assumed to be in XML format (content-type "text/xml").

**Note:** Use the HTTP “Content-Type” header to specify a different content type. To specify that content is gzipped, use the HTTP “Content-Encoding” header.

### General URL syntax

To invoke an operation via HTTPS POST, submit a URL that has the following general syntax:

`https://`*`host`*`[/itmind/mind`*`version`*`]`

where:

- *`host`* is one of the following host names:
  - QA server: `otwsqa.tivo.com`
  - Production server: `otws.tivo.com`
- *`version`* is the web service version number, for example, 9. Use the optional path element following the host name to consume a web service version different than the default (version 8). For more information on using a specific web service version, see 3.2.3 *Web Services Versioning*, below.

### Example

For example, here’s a simple XML document that if POSTed invokes the `customerInfoSearch` operation from the `accountservice` service. This operation takes a single `partnerCustomerId` string as input.:

```
<customerInfoSearch>
  <partnerCustomerId>0900000003</partnerCustomerId>
</customerInfoSearch>
```

### Nested sibling elements

When an XML document contains nested sibling elements (children of another element), POST requires that sibling elements in XML documents be listed in alphabetical order. For example, in the following code the `customer`, `device`, and `service` elements are siblings (children) of the `contract` element, so they are listed in alphabetical order. Similarly, because the `city`, `country`, `postalCode`, `state`, and `street1` elements are siblings of the `address` element, they are also listed in alphabetical order.

```
<serviceActivate>
  <contract>
    <customer>
      <address>
        <city>Alviso</city>
        <country>US</country>
        <postalCode>95002</postalCode>
        <state>CA</state>
        <street1>2160 gold st</street1>
      </address>
      <email>jmarada@tivo.com</email>
      <firstName>Jaya</firstName>
      <lastName>Marada</lastName>
      <partnerCustomerId>234234</partnerCustomerId>
      <phone>4089991234</phone>
    </customer>
    <device>
      <tivoSerialNumber>74683742372741243</tivoSerialNumber>
    </device>
    <service>
      <sku>S00055</sku>
    </service>
  </contract>
</serviceActivate>
```

## 3.2.2 Using GET

An HTTPS GET request passes parameters in the body of a URL.

### General URL syntax

To invoke an operation via HTTPS GET, submit a URL that has the following general syntax:

`https://`*`host`*`[/itmind/mind`*`version`*`]/`
*`operation`*`?type=`*`operation`*`&`*`path.to.param`*`=`*`value`*`[&`*`path.to.param`*`]=`*`value`*

where:

- *`host`* is one of the following host names:
  - QA server: `otwsqa.tivo.com`
  - Production server: `otws.tivo.com`
- *`version`* is the web service version number, for example, 9. Use the optional path element following the host name to consume a web service version different than the default (version 8). For more information on using a specific web service version, see 3.2.3 *Web Services Versioning*, below.
- *`operation`* specifies an operation, for example, `customerInfoSearch`.

**Note:** Partners must integrate with the QA web services and verify test results before invoking production services.

### Example

For example, here is a simple GET that invokes the `customerInfoSearch` operation from the `accountservice` service. This operation takes a single `partnerCustomerId` string as input

```
https://otwsqa.tivo.com/customerInfoSearch?type=customerInfoSearch
&partnerCustomerId=0900000003
```

The long line in this URL example wraps to a second line.Remember that code listings and sample URLs throughout this document may wrap long lines, and may have spaces inserted in them to make the wrapped lines easier to read. If you copy and paste code examples and URLs you may have to edit them to make them work as expected. For example, you may have to delete extraneous spaces from a URL.

## 3.2.3 Web Services Versioning

TiVo web services support versioning. To consume a version of the web services other than the default (version 8), submit either a GET or a POST request with an additional path element (`itmind/mind`*`version`* where *version* is the version number) following the host name. Note the host names are unchanged: `otwsqa.tivo.com` for QA server, `otws.tivo.com` for production server.

Thus the general syntax to consume a particular version of the TiVo web services is:

```
https://host/itmind/mindversion
?type=operation
&path.to.param=value[&path.to.param]=value
```

Note: The above URL is shown on multiple lines for clarity.

Version 9 is the most current version of the web services. For example, to call `configInfoSearch` version 9, use the following URL:

```
https://otwsqa.tivo.com/itmind/mind9
?type=configInfoSearch&tivoSerialNumber=746000190230D38
```

The addition of `/itmind/mind9` indicates to use version 9 of the web services.

You can invoke default web service version 8 operations using either the explicit versioned URL:

```
https://host/itmind/mind8
?type=operation
&path.to.param=value[&path.to.param]=value
```

Or the default “unversioned” URL:

```
https://host/operation
?type=operation
&path.to.param=value[&path.to.param]=value
```

### 3.2.4 Specifying Parameter Values in a URL

When a parameter's only ancestor is the operation, the *path.to.param* is simply the parameter name. For example, in the following XML listing, the only ancestor of the parameter `partnerCustomerId` is the operation `customerInfoSearch`

```
<customerInfoSearch>
  <partnerCustomerId>0900000003<partnerCustomerId>
</customerInfoSearch>
```

Therefore, the corresponding URL for an HTTPS GET request looks like this:

```
https://otws.tivo.com/customerInfoSearch?type=customerInfoSearch
&partnerCustomerId=0900000003
```

When a parameter has one or more ancestors between it and the operation, the *path.to.param* lists the name of each ancestor of the parameter followed by a dot, followed by an integer (where 0 specifies the first instance of an ancestor), for example, `customer.0.customerId`.

Here's an example of an XML document that if POSTed invokes the `serviceActivate` operation from the `commerceservice` service:

```
<serviceActivate>
  <contract>
    <customer>
      <address>
        <city>Alviso</city>
        <country>US</country>
        <postalCode>95002</postalCode>
        <state>CA</state>
        <street1>2160 gold st</street1>
      </address>
      <email>jmarada@tivo.com</email>
      <firstName>Jaya</firstName>
      <lastName>Marada</lastName>
      <partnerCustomerId>234234</partnerCustomerId>
      <phone>4089991234</phone>
    </customer>
    <device>
      <tivoSerialNumber>74683742372741243</tivoSerialNumber>
    </device>
    <service>
      <sku>S00055</sku>
    </service>
  </contract>
</serviceActivate>
```

Here is the corresponding URL, which calls the `commerceservice` service to invoke the `serviceActivate` operation:

```
https://otpwsqa.tivo.com/serviceActivate?type=serviceActivate
&contract.0.customer.0.partnerCustomerId=234234
&contract.0.customer.0.phone=4089991234
```

```
&contract.0.customer.0.email=jmarada@tivo.com
&contract.0.customer.0.firstName=Jaya
&contract.0.customer.0.lastName=Marada
&contract.0.customer.0.address.0.street1=2160+Gold+St
&contract.0.customer.0.address.0.street2=apt+123
&contract.0.customer.0.address.0.city=Alviso
&contract.0.customer.0.address.0.state=CA
&contract.0.customer.0.address.0.postalCode=95002
&contract.0.customer.0.address.0.country=US
&contract.0.device.0.tivoSerialNumber=74683742372741243
&contract.0.service.0.sku=S00055
```

In the URL above, the syntax fragment that specifies the path to the value of the `partnerCustomerId` parameter is:

`contract.0.customer.0.partnerCustomerId=234234`

This syntax indicates that the `partnerCustomerId` parameter has two ancestors, `contract` and `customer`, and that particular customer is the first (and in this case, the only) such instance in the document.Following are some examples of XML documents and the corresponding URLs. These do not invoke actual web service operations; their purpose is to demonstrate the rules for constructing web service URLs based on XML documents.

## Example 1

In this example, a service operation named `myOp` has parameters `abc` and `xyz`, which have the same ancestor, `foo`. The XML document contains only one instance of `foo`.

| | |
|---|---|
| XML | <myOp><br><foo><br><abc>DOG</abc><br><xyz>CAT</xyz><br></foo><br></myOp> |
| URL | https://*host*/myOp?type=myOp&foo.0.abc=DOG&foo.0.xyz=CAT |

## Example 2

In this example, a service operation named `myOp` has the parameter `abc`, which has two ancestors, `foo` and `bar`.

| | |
|---|---|
| XML | <myOp><br><foo><br><bar><br><abc>FISH</abc><br></bar><br></foo><br></myOp> |

| URL | https://*host*/myOp?type=myOp&foo.0.bar.0.abc=FISH |
|---|---|

### Example 3

In this example, a service operation named `myOp` has the parameters `abc` and `xyz`, which have the same ancestor, `foo`. The XML document contains two instances of each parameter, and two instances of `foo`.

**Note:** The URL example wraps on a printed page. In actual usage, the URL would submit the request as a single line.

| XML | <myOp><br>  <foo><br>  <abc>DOG</abc><br>  <xyz>CAT</xyz><br>  </foo><br>  <foo><br>  <abc>PIG</abc><br>  <xyz>HEN</xyz><br>  </foo><br></myOp> |
|---|---|
| URL | https://*host*/myOp?type=myOp<br>&foo.0.abc=DOG<br>&foo.0.xyz=CAT<br>&foo.1.abc=PIG<br>&foo.1.xyz=HEN |

## 3.3 Receiving a Response

TiVo partner web service operations return an XML response document containing data specific to the particular operation. Elements in the XML response document are in alphabetical order. If a field type definition is a structure, it will include the `type` element. If the request fails, the document also contains an `error` element.

**Note:** Some operations return an `appError` element instead of an `error` element. The structures of the `appError` and `error` elements are identical, and the information they return is also identical. For more information about `appError`, see *appError* on page 90. For more infromation about `error`, see *error* on page 117.

The following listing shows a `serviceActivateResponse` document returned by a successful `serviceActivate` operation. The `contract` structure includes a `type` element. This response does not contain an `error` element.

```
<serviceActivateResponse>
  <contract>
    <customer>
      <id>0012345678</id>
```

```
      <levelOfDetail>high</levelOfDetail>
      <type>customer</type>
    </customer>
    <id>987654321</id>
    <levelOfDetail>high</levelOfDetail>
    <type>contract</type>
  </contract>
  <status>success</status>
  <requestId>6f5fa0fc-9802-44e0aece455c4d4b9a90</requestId>
</serviceActivateResponse>
```

The following listing shows a document returned by an unsuccessful `serviceActivate` operation. This response contains an `error` element that describes the problem.

```
<error>
  <code>1000</code>
  <text>Invalid TiVo Serial Number. TiVo Serial Number should be a
     15-digit alpha numeric.</text>
  <requestId>55b91685-178a-43e5-8169-adcbae810cb5</requestId>
</error>
```

An error element has the following structure:

| Field Name | Type | Description |
|---|---|---|
| code | string | Error code. Error codes and error messages can be generated by processes at various levels within the TiVo services system. |
| text | string | Text description of the error. |
| requestId | string | A unique request ID. |

## 3.4 Common Tasks

Here's a list of some common tasks, the corresponding Web services operations, and the service group to which each operation belongs.

| Task | Operation | Service Group |
|---|---|---|
| Retrieve customer information | customerInfoSearch | Account service |
| Activate service | serviceActivate | Commerce service |
| Turn off (cancel) active service | serviceCancel | Commerce service |
| Reset or initialize a device | serviceReset | Commerce service |
| Retrieve information about a specified device | deviceInfoSearch | Device service |
| Update device attributes, including the name of the device | deviceInfoStore | Device service |
| Schedule a recording | scheduleRecordingStore | Online service |

# The TiVo Service Data Model

4

This section describes the data model used by TiVo to manage TiVo service to devices. The key data model entities and key attributes are described in this table:

| Entity | Unique identifiers | Key attributes |
|---|---|---|
| Customer | Customer email, `partnerCustomerID` (if used by partner), or `tivoCustomerId` | Privacy level (opt status), media access key (MAK) |
| Device | TiVo Serial Number and Hardware Serial Number (if used by partner) | Service state, device name, feature entitlement, feature configuration, service groups |

## 4.1 Customer

A customer represents a household, which can have one or more TiVo devices activated. All devices activated on the same customer and connected within the same home network can interoperate using TiVo multi-room features for which the device is entitled. (See "Feature Entitlement and Configuration").

### 4.1.1 Unique Customer Identifier

Customers are identified in TiVo's system with one or more unique keys:

- Customer email: For partners who allow their customers to use tivo.com, customer email is the customer login.
- `partnerCustomerID`: This is a unique identifier that is provided to TiVo by the partner and usually corresponds to a unique customer identifier in the partner billing system

The partner can specify the customer email (not the preferred method), the `partnerCustomerID`, or `tivoCustomerId` in web service calls to identify the customer. For US partners who want to allow multi-room features between partner-managed devices and retail devices, the partner must specify the customer email for the customer's retail TiVo account in order to add the partner-managed device to the customer.

### 4.1.2 Privacy Level

The TiVo Service determines the data on a device that can be shared with the TiVo Service based on the privacy level assigned to the customer. This information can include information that is retrieved by web service calls, data that is used to provide features to the user, and audience research information. Prior to integrating with the partner web services, the partner specifies the default privacy level to which all new customers created by that partner should be set. The

`customerInfoStore` operation allows some partners to change this privacy level. The `optStatus` field represents the customer privacy level.

Valid privacy levels are `optOut`, `optNeutral`, and `optIn`. See *optStatus* on page 124 for more information.

Most partners choose to have new customers set up with the `optNeutral` privacy level.

See TiVo's privacy policy for more information:
http://www.tivo.com/abouttivo/policies/tivoprivacypolicy.html

### 4.1.3 Media Access Key

The Media Access Key (MAK) is used by applications connected to the home network that need to access a TiVo device. For example, both the TiVo Desktop PC software and the TiVo iPad application require a user enter the device Media Access Key during setup. The Media Access Key is created at the time of customer creation, and is assigned to each active device activated for a customer. This means features that require the Media Access Key are not available prior to activation.

**Note:** The MAK key is assigned at the customer level and not per device.

## 4.2 About the Device Entity

A device entity represents the service data configured on a client device (DVR, etc.) by the TiVo service.

At manufacturing time, each device is assigned a unique TiVo Serial Number and a Hardware Serial Number, if different from the TiVo Serial Number. The TiVo Service Number (TSN) must contain exactly 15 hexadecimal characters [a-f, A-F, 0-9]. You can associate a TiVo customer with one or more devices, and a device with more than one customer record. A device in its active state, however, can be associated only with a single customer. For example, you might have originally activated a device with customer A, but then canceled and re-activated it with customer B. The active device is therefore associated only with customer B. Therefore it is important to parse the device service state when retrieving the devices associated with a customer account.

### 4.2.1 Serial Number

Each TiVo device is assigned a TiVo Serial Number (TSN) at the time of device manufacturing or refurbishment. Most partners use the TSN as the device identifier for partner web service calls. In some rare cases the partner may already have a different serial number assigned to that same device. In that case, the partner can choose to use the Hardware Serial Number in web service calls instead. In that model, TiVo would convert the Hardware Serial Number to the TiVo Serial Number for use in all of TiVo's systems.

**Note:** Using a Hardware Serial Number as a TSN adds latency to each web service call due to the lookup process.

## 4.2.2 Service State

Each TiVo device is assigned a service state that changes during the device life cycle. For example, when a device is manufactured, it is initialized to a service state of `New`. When the TiVo service is activated for a limited service term (monthly or yearly, for instance), the service state is set to `Good`. When the TiVo service is activated for lifetime service, the service state is set to `Product Lifetime`.

Partners can use the following services to manipulate device service states:

- Use the Commerce Service to change the service state of a customer's device. (The `serviceActivate`, `serviceCancel`, and `serviceReset` operations, which change the service state of a device, are in the Commerce Service.) When the Commerce Service is initiated, the new service state is available to the device within 15 minutes. The device will receive its updated service state after it makes a service connection.
- Use the Account Service to retrieve the current service state of a device.

The following table defines valid service states for a TiVo device:

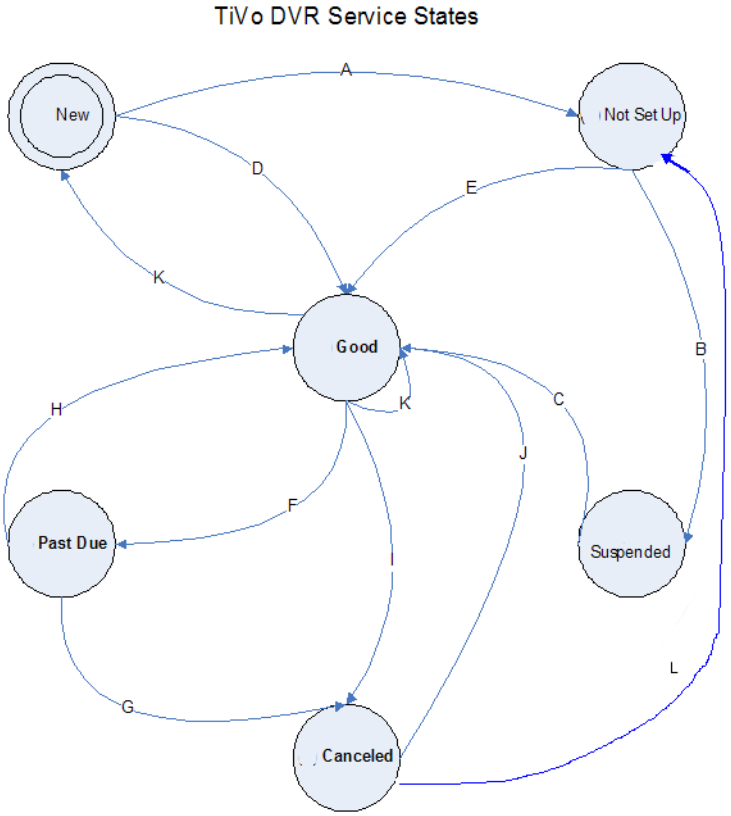
| State Number | Type | Description |
|---|---|---|
| 1 | New | Initial service state of a device after manufacturing. |
| 2 | Not Set Up | Service assigns this to device after the device has completed guided setup. The device must be activated within a mutually agreed upon grace period or it will be moved to Suspended state. |
| 3 | Good | Device is in good standing (activated for TiVo service). |
| 6 | Past Due | Partner determines customer account is past due and would like device messaging to appear requesting customer contact partner to return account into good standing. Not applicable for all partners. Check with your TiVo account manager for details. Device continues to receive guide data and all features remain enabled. |
| 7 | Suspended | If device remains in service state Not Set Up past the grace period, the device is automatically moved to the Suspended service state. It will no longer receive guide data, and most TiVo functionality will be disabled. |
| 8 | Canceled or Closed | Customer cancels service for this device with partner. Once in this service state, device will no longer receive guide data and most TiVo functionality will be disabled. |

**Note:** Service states are most often referred to by their service state numbers.

### Service State Transitions

The service state of the TiVo device changes during the device life cycle. For example, a device's service state may change from `Good` to `Past Due` and then return to `Good`.

The following diagram shows allowed state transitions for a TiVo device. The states Good, Past Due, and Canceled can be controlled via the commerce service.




The following table describes use cases for each transition.

| Transition | Initiated By | Description |
|---|---|---|
| A | TiVo Service | When customer completes guided setup on device, the TiVo service moves the device into service state Not Set Up. |
| B | TiVo Service | If device has not been activated at the end of its grace period in service state Not Setup, TiVo service moves it into service state Suspended. |
| C | serviceActivate | Customer contacts partner to activate service for a device. |
| D | serviceActivate | Partner can activate a device any time after the device is manufactured, if the partner knows to which customer a specific device will be sent. This use case is how TiVo pre-activates devices purchased from the TiVo.com online store. TiVo activates the device once it receives shipping confirmation for a customer order (including the TSN of the device which was shipped to that customer). |
| E | serviceActivate | Customer contacts partner to activate. |
| F | serviceCancel | (Only applicable for some partners. Check with your TiVo account manager). Partner identifies customer account is delinquent and wants device to message customer to contact partner. |

| Transition | Initiated By | Description |
|---|---|---|
| G | serviceCancel | Partner cannot resolve delinquent account with customer and chooses to cancel service on that device. If customer has multiple devices with partner, and partner wishes to cancel service for all of them, partner must invoke the cancel service to cancel each device individually. |
| H | serviceActivate | Partner and customer reach terms on delinquent account, and partner re-activates account. |
| I | serviceCancel | Customer contacts partner to cancel service on device. Alternately, if partner is selling service to customers with explicit end dates, partner could chose to cancel service when partner billing system identifies service end date has been reached. |
| J | serviceActivate | Customer contacts partner to re-activate service. Alternately, previous customer may have sold or given away device to new customer who contacts partner to activate. |
| K | serviceCancel, serviceActivate | Customer has received new device and wishes to transfer service from an existing device to a new device. Alternately, partner may have shipped replacement device to customer as part of RMA process and wishes to transfer service to the replacement unit. New unit takes on same service state as old device. Old unit will be placed into service state New in the device system information screen. |
| L | serviceReset | Partner initializes the device and resets the device settings after cancelling service. |
| Not shown | Move device from one customer account to another | This is different from transferring service. Partner cancels service on a device for customer 1, then activates service for that device for customer 2. |

### 4.2.3 Device Name

Each TiVo device is assigned a default name that is displayed on the device System & Account Information Screen and My Shows List (when using multi-room features), as well as on tivo.com (for partners whose customers use tivo.com) and networked applications including TiVo Desktop and the TiVo iPad application. The default name for a device is:

- <last 4 digits of TSN> DVR (for TiVo DVR devices)

The partner may change this name via the `deviceInfoStore` operation. Most users choose to rename their devices based on location (example: "Living Room", "Master Bedroom") and so forth.

### 4.2.4 Feature Entitlement and Configuration

The TiVo Service provides each device its own unique, secured specification of features that the device is entitled to use. Some features require additional configuration and these configuration parameters are also delivered to the device in the same manner.

During the partner integration process, your TiVo account manager collects information that TiVo personnel use to specify the features to which your customers are entitled. For some features, TiVo also need to collect configuration parameters required for specific partner features (for example, Video On Demand). TiVo sets up the provisioning system is entitle the correct set of features for new

partner devices by default. It is also set up to assign defaults for features that require additional configuration. The partners can use the `deviceSettingsStore` operation to override these default entitlements at any time, as well as overriding some configuration settings.

The following table lists the current set of TiVo feature entitlements that can be controlled by partner web services:

| TiVo Feature | Entitlement Control Field | Application Notes & Examples |
|---|---|---|
| Download broadband content | cds | TiVoCast downloads, Amazon content downloads, YouTube |
| Ability to connect to and use an external hard drive | esata | |
| Access broadband applications | hme | TiVo personnel manually publish and configure the applications which a device will be configured for by default and this cannot be overridden via partner web services. See “device service groups” for information on how the partner can add or remove additional applications |
| Allow HTTP access to the device | http | |
| Transfer content from one DVR device to another DVR device | mrv | Multi-room feature. Available on series 2, series 3 and series 4 devices |
| Stream content from one DVR device to another device | mrs | Multi-room feature. Coming soon. To be soon available on series 4 devices. |
| Share music and photos from the PC and enable the Music & Photos menu item in the TiVo Classic (SD) User Interface | musicphoto | |
| Allow recordings to be scheduled remotely | tco | |
| Transfer content from the TiVo Desktop PC application (or for the Internet via TiVo Desktop) to a DVR device | ttcb | |
| Transfer content from a DVR device to a PC using TiVo Desktop | ttg | |

The following table describes features that require special configuration by TiVo and the partner. Please see your TiVo Account Manager or project Statement of Work to determine if any of the following are applicable to your integration:

| Partner Feature | Entitlement Control Field | Configuration Control Field | Application Notes & Examples |
|---|---|---|---|
| Video on Demand | vod | SiteID | Specifies which VOD server and catalog (if applicable) the device will access (view content, search) |

| Partner Feature | Entitlement Control Field | Configuration Control Field | Application Notes & Examples |
|---|---|---|---|
| Impulse Pay Per View | ppv | msoServiceID | The msoServiceID is provided by the partner and is used to determine specific configuration settings for partner features including impulse Pay Per View, Switched Digital Video and Virtual Private Network. |
| Switched Digital Video | sdv | msoServiceID | See application note for Impulse Pay Per View, above. |
| Virtual Private Network | vpn | msoServiceID | See application note for Impulse Pay Per View, above. |
| Caller ID | callerID | N/A | |

**Note:** Actual values for the `siteID` and `msoServiceID` are defined during the integration process. Partner features (for example, VOD server/catalog metadata extraction) are configured by partner and TiVo personnel.

**Note:** In the future some partner features may be configured directly by the TiVo Service during an autodiscovery process. If that is the case for your integration, TiVo configures the TiVo Service to allow the `msoServiceID` and `siteID` to be discovered by the service. You can use the `overrideAutoDiscovery` field to force a setting made by the partner web service to remain in place regardless of the value the service discovered.

### 4.2.5 Distribution Groups

TiVo uses distribution groups (DGs) to define populations of devices that are to receive the same configuration for a specific feature or set of features. Distribution groups whose membership is automatically managed by the TiVo service are called default distribution groups.

Partner might want to offer a feature to a subset of their population. They can use the service group management operations such as `serviceGroupStore` and `serviceGroupRemove` to manage the subset population (device serial number list). This requires additional work by TiVo personnel to configure a custom service group for the desired feature set. See your TiVo Account manager for more information.

### 4.2.6 Dial Code Groups

Dial code groups specify which devices connect to a particular TiVo Service Center. All production devices are automatically configured to connect to the production service center, but some partners (mainly International partners who have their own TiVo Service Center) use the `dialCodeUpdate` operation of the device service to move a device to a different service center.

## 4.3 When Do Devices Get New Data?

Generally service data changes made by the partner web services are available to the TiVo Service within 15 minutes. The device receives the updated service data during its next service connection. A user can also choose to force a service connection from a TiVo device.

# Account Service

5

The Account service provides operations for working with customer account data. The data structures used by these operations are defined in *XML Objects* on page 89.

## 5.1 Operations

The following table summarizes the `Account` service operations:

| Operation Name | Description | Input | Output |
|---|---|---|---|
| customerInfoSearch | Get customer information and contract information. | customerRequest Field Group<br>contractStatus - Optional | customerInfoSearchResponse |
| customerInfoStore | Creates or updates customer information. | customerStoreRequest Field Group | customerInfoStoreResponse |
| resetPassword | Resets the password of the customer.<br>NOTE: This operation is not available by default to all partners. | Email address | resetPasswordResponse element |

## 5.2 customerInfoSearch Operation

Use the `customerInfoSearch` operation to retrieve a specified customer's account information, including the customer's device list, media access key, and privacy status.

### 5.2.1 Input Parameters

This operation requires a `customerRequest` field group as input. The `customerRequest` field group can be one or more of the following:

- `partnerCustomerId`
- `tivoCustomerId`
- `tivoSerialNumber`
- `hardwareSerialNumber`

At least one of these is required. The operation automatically filters result using the fields provided in the request. For example, if the request specifies `tivoCustomerId` and `tivoSerialNumber` (TSN), the return result only includes the customer and the contract associated with the given TSN.

The operation also accepts an optional parameter, `contractStatus`, that further filters the result by the customer's contract status. The `contractStatus` parameter can be one of the following values:

- A, to show only *active* contracts.
- C, to show only *canceled* contracts.

If the `contractStatus` parameter is not specified or left blank, the `customerInfoSearch` operation returns all active and canceled contracts.

## Sample XML Input

The following listings show sample XML input for four `customerInfoSearch` operations, each of which specifies one of the different field values for `customerRequest` field group:

```
<customerInfoSearch>
  <tivoCustomerId>0011955809</tivoCustomerId>
</customerInfoSearch>

<customerInfoSearch>
  <partnerCustomerId>0900000003</partnerCustomerId>
</customerInfoSearch>

<customerInfoSearch>
  <tivoSerialNumber>74683742372741243</tivoSerialNumber>
</customerInfoSearch>

<customerInfoSearch>
  <hardwareSerialNumber>C01234567890123</hardwareSerialNumber>
</customerInfoSearch>
```

## Sample URL Request

The following URLs demonstrate how to invoke the `customerInfoSearch` operation with different `customerRequest` field group parameters.

```
https://otwsqa.tivo.com/customerInfoSearch?
type=customerInfoSearch&tivoCustomerId=0011955809

https://otwsqa.tivo.com/customerInfoSearch?
type=customerInfoSearch&partnerCustomerId=0900000003

https://otwsqa.tivo.com/customerInfoSearch?
type=customerInfoSearch&tivoSerialNumber=74683742372741243

https://otwsqa.tivo.com/customerInfoSearch?
type=customerInfoSearch&hardwareSerialNumber=C01234567890123

https://otwsqa.tivo.com/customerInfoSearch?type=customerInfoSearch&
hardwareSerialNumber=C01234567890123&contractStatus=A
```

## 5.2.2 Output Element

The `customerInfoSearch` operation returns either a `customerInfoSearchResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The following listing shows a `customerInfoSearchResponse` for a successful `customerInfoSearch` operation:

```
<customerInfoSearchResponse>
  <customerList>
    <customer>
      <address>
        <city>ALVISO</city>
        <country>US</country>
        <levelOfDetail>high</levelOfDetail>
        <postalCode>95002</postalCode>
        <state>CA</state>
        <street1>216 silver street</street1>
        <street2 />
        <type>address</type>
      </address>
      <customerContractList>
        <contract>
          <device>
            <hardwareSerialNumber>C80860227XBFDSRK
              </hardwareSerialNumber>
            <levelOfDetail>high</levelOfDetail>
            <partnerId>0000003721</partnerId>
            <subPartnerId />
            <tivoSerialNumber>CF0010E2D05DE04</tivoSerialNumber>
            <type>device</type>
          </device>
          <endDate>2010-07-21</endDate>
          <id>0107493991</id>
          <levelOfDetail>high</levelOfDetail>
          <service>
            <description>MSO Monthly Service</description>
            <levelOfDetail>high</levelOfDetail>
            <sku>S00055</sku>
            <type>service</type>
          </service>
          <startDate>2010-07-21</startDate>
          <status>Cancelled</status>
          <type>contract</type>
        </contract>
        <contract>
```

```
      <device>
        <hardwareSerialNumber>C80860227XBFDSRK
          </hardwareSerialNumber>
        <levelOfDetail>high</levelOfDetail>
        <partnerId>0000003721</partnerId>
        <subPartnerId />
        <tivoSerialNumber>CF0010E2D05DE04</tivoSerialNumber>
        <type>device</type>
      </device>
      <endDate>2010-07-21</endDate>
      <id>0107493988</id>
      <levelOfDetail>high</levelOfDetail>
      <service>
        <description>MSO Monthly Service</description>
        <levelOfDetail>high</levelOfDetail>
        <sku>S00055</sku>
        <type>service</type>
      </service>
      <startDate>2010-07-21</startDate>
      <status>Cancelled</status>
      <type>contract</type>
    </contract>
    <contract>
      <device>
        <hardwareSerialNumber>C80860227XBFDSRK
          </hardwareSerialNumber>
        <levelOfDetail>high</levelOfDetail>
        <partnerId>0000003721</partnerId>
        <subPartnerId />
        <tivoSerialNumber>CF0010E2D05DE04</tivoSerialNumber>
        <type>device</type>
      </device>
      <endDate>2010-07-21</endDate>
      <id>0107493985</id>
      <levelOfDetail>high</levelOfDetail>
      <service>
        <description>MSO Monthly Service</description>
        <levelOfDetail>high</levelOfDetail>
        <sku>S00055</sku>
        <type>service</type>
      </service>
      <startDate>2010-07-21</startDate>
      <status>Cancelled</status>
      <type>contract</type>
    </contract>
    <type>contractList</type>
  </customerContractList>
  <email>PartnerTester01@partner.com</email>
```

```
    <firstName>Pat</firstName>
    <id>0012093599</id>
    <lastName>Tester1</lastName>
    <levelOfDetail>high</levelOfDetail>
    <optStatus>optIn</optStatus>
    <partnerCustomerId>PartnerTester01</partnerCustomerId>
    <partnerId>0000003721</partnerId>
    <phone>1234567890</phone>
    <type>customer</type>
  </customer>
  <customer>
    <address>
      <city>ALVISO</city>
      <country>US</country>
      <levelOfDetail>high</levelOfDetail>
      <postalCode>95002</postalCode>
      <state>CA</state>
      <street1>243 bron street</street1>
      <street2 />
      <type>address</type>
    </address>
    <customerContractList>
      <contract>
        <device>
          <hardwareSerialNumber>C80860227XBFDSRK
            </hardwareSerialNumber>
          <levelOfDetail>high</levelOfDetail>
          <partnerId>0000003721</partnerId>
          <subPartnerId />
          <tivoSerialNumber>CF0010E2D05DE04</tivoSerialNumber>
          <type>device</type>
        </device>
        <id>0107494028</id>
        <levelOfDetail>high</levelOfDetail>
        <service>
          <description>MSO Monthly Service</description>
          <levelOfDetail>high</levelOfDetail>
          <sku>S00055</sku>
          type>service</type>
        </service>
        <startDate>2010-07-21</startDate>
        <status>Active</status>
        <type>contract</type>
      </contract>
      <contract>
        <device>
          <hardwareSerialNumber>C80860227XBFDSRK
            </hardwareSerialNumber>
```

```
        <levelOfDetail>high</levelOfDetail>
        <partnerId>0000003721</partnerId>
        <subPartnerId />
        <tivoSerialNumber>CF0010E2D05DE04</tivoSerialNumber>
        <type>device</type>
      </device>
      <endDate>2010-07-21</endDate>
      <id>0107493989</id>
      <levelOfDetail>high</levelOfDetail>
      <service>
        <description>MSO Monthly Service</description>
        <levelOfDetail>high</levelOfDetail>
        <sku>S00055</sku>
        <type>service</type>
      </service>
      <startDate>2010-07-21</startDate>
      <status>Cancelled</status>
      <type>contract</type>
    </contract>
    <contract>
      <device>
        <hardwareSerialNumber>C80860227XBFDSRK
          </hardwareSerialNumber>
        <levelOfDetail>high</levelOfDetail>
        <partnerId>0000003721</partnerId>
        <subPartnerId />
        <tivoSerialNumber>CF0010E2D05DE04</tivoSerialNumber>
        <type>device</type>
      </device>
      <endDate>2010-07-21</endDate>
      <id>0107493987</id>
      <levelOfDetail>high</levelOfDetail>
      <service>
        <description>MSO Monthly Service</description>
        <levelOfDetail>high</levelOfDetail>
        <sku>S00055</sku>
        <type>service</type>
      </service>
      <startDate>2010-07-21</startDate>
      <status>Cancelled</status>
      <type>contract</type>
    </contract>
    <type>contractList</type>
  </customerContractList>
  <email>PartnerTester02@partner.com</email>
  <firstName>Partner</firstName>
  <id>0012093601</id>
  <lastName>tester2</lastName>
```

```
      <levelOfDetail>high</levelOfDetail>
      <partnerCustomerId>partnertester01</partnerCustomerId>
      <partnerId>0000003721</partnerId>
      <phone>0987654321</phone>
      <optStatus>optIn</optStatus>
      <type>customer</type>
    </customer>
    <type>customerList</type>
  </customerList>
  <requestId>b1eb2450-c86a-4132-9186-b5cd877f3e7f</requestId>
  <status>success</status>
</customerInfoSearchResponse>
```

The following listing shows a `customerInfoSearchResponse` for an unsuccessful `customerInfoSearch` operation. When an error occurs, the `customerInfoSearchResponse` consists only of the `error` element:

```
<error>
  <code>1011</code>
  <text>Hardware Serial Number not found</text>
  <requestId>1234567890-abc-1234567890</requestId>
</error>
```

# 5.3 customerInfoStore Operation

The `customerInfoStore` operation creates a new customer record or make updates to an existing customer record. If you call this operation with a non-existent customer ID, then it will create a new customer record; if you call it with an existing customer ID, then it updates that customer record.

**Note:** Use this operation to modify the current `partnerCustomerId` associated with an existing TiVo customer through the `newParterCustomerId` field. You can use this field to change the `partnerCustomerId` of an existing customer to a new `partnerCustomerId` that does not already exist in the TiVo system.

## 5.3.1 Input Parameter

The `customerInfoStore` operation requires a `customer` element as input. For specific information about the `customer` element, see *customer* on page 106. The following examples, using the same customer data, illustrate updating an existing customer record using XML as input to a POST operation, and passing parameters in a URL for a GET operation.

### Sample XML Input

The following listing shows sample XML input for a `customerInfoStore` operation.

```
<customerInfoStore>
  <customer>
    <address>
      <city>alviso</city>
```

```
      <country>US</country>
      <postalCode>95002</postalCode>
      <state>ca</state>
      <street1>2160 gold st</street1>
    </address>
    <firstName>Mary</firstName>
    id>1234</id>
    lastName>Sasomsub</lastName>
    newPartnerCustomerId>234234</newPartnerCustomerId>
    <optStatus>optIn</optStatus>
    <partnerCustomerId>234234</partnerCustomerId>
    <phone>4085559468</phone>
  </customer>
  <newPartnerCustomerId>234234</newPartnerCustomerId
</customerInfoStore>
```

### Sample URL Request

The following URL invokes the `customerInfoStore` operation.

```
https://otwsqa.tivo.com/customerInfoStore?type=customerInfoStore
&customer.0.id=1234
&customer.0.firstName=Mary
&customer.0.lastName=Sasomsub
&customer.0.address.0.street1=2160+gold+st
&customer.0.address.0.city=alviso
&customer.0.address.0.state=ca
&customer.0.address.0.postalCode=95002
&customer.0.address.0.country=us
&customer.0.newPartnerCustomerId=234234
&customer.0.partnerCustomerId=1234
&customer.0.optStatus=optIn
&newPartnerCustomerId=234234
```

## 5.3.2 Output Element

The `customerInfoStore` operation returns either a `customerInfoStoreResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The following listing shows a `customerInfoStoreResponse` for a successful `customerInfoStore` operation.

```
<customerInfoStoreResponse>
  <requestId>550984335826794057 5</requestId>
  <status>success</status>
</customerInfoStoreResponse>
```

# 5.4 resetPassword

Resets the password of the customer. It sends the temporary password to the provided email in the request.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

## 5.4.1 Input Parameters

This operation requires an email as input. This email is the customer identifier email.

### Sample XML Input

The following listing shows sample XML input for a `resetPassword` operation.

```
<resetPassword>
  <email>email@tivo.com</email>
</resetPassword>
```

### Sample URL Request

The following URL invokes the `resetPassword` operation.

```
https://otwsqa.tivo.com/
  resetPassword?type=resetPassword&email=email@tivo.com
```

## 5.4.2 Output Element

The `resetPassword` operation returns either a `resetPasswordResponse` element or an error element. Only if the operation triggers an error does the response consist of an error element. The `resetPasswordResponse` element contains either the success or failure status. If the password is successfully reset and sends an email to the given email in the request, the elemet contains the success response. Otherwise, contains the failure response.

### Sample XML Response

The following listing shows a resetPasswordResponse for a successful `resetPassword` operation.

```
<resetPasswordResponse>
  <requestId>5509843358267940575</requestId>
  <status>success</status>
</resetPasswordResponse>
```

# Commerce Service

6

The Commerce service provides operations that activate service on a device, cancel service on a device, and reset a device. The data structures used by these operations are defined in *XML Objects* on page 89.

## 6.1 Operations

The following table summarizes the Commerce service operations:

| Operation Name | Description | Input | Output |
|---|---|---|---|
| serviceActivate | Activate a device. | activationFieldGroup | serviceActivateResponse |
| serviceCancel | Cancel a service to a device. | serviceCancelFields | serviceCancelResponse |
| serviceCancelbyCustomerId | Cancels all active contracts under a specific customer account. | serviceCancelByCustomerId Fields | serviceCancelByCustomerId ResponseList |
| serviceReset | Reset flag on a device. | serviceResetFieldGroup | serviceResetResponse |

## 6.2 serviceActivate Operation

Use the `serviceActivate` operation to activate a device and make the TiVo service available to it. This operation also creates a customer record if this is the first time the customer has activated a device.

### 6.2.1 Input Parameter

The `serviceActivate` operation requires an `activationFieldGroup` element as an input parameter. The `activationFieldGroup` element wraps a `contract` element that describes the services being activated. For more information about the `activationFieldGroup`, see *activation Field Group* on page 89. For more information about the `contract` element, see *contract* on page 103.

#### Sample XML Input

The following listing shows a sample XML input for a `serviceActivate` operation.

```
<serviceActivate>
  <contract>
    <customer>
      <address>
        <city>Alviso</city>
```

```
        <country>US</country>
        <postalCode>95002</postalCode>
        <state>CA</state>
        <street1>2160 gold st</street1>
      </address>
    <email>jmarada@example.com</email>
    <firstName>Jaya</firstName>
    <lastName>Marada</lastName>
    <partnerCustomerId>234234</partnerCustomerId>
    <phone>4085199468</phone>
    </customer>
    <device>
      <hardwareSerialNumber>746321323213214</hardwareSerialNumber>
      <msoServiceId>1</msoServiceId>
      <siteId>36</siteId>
    </device>
    <service>
      <sku>S00055</sku>
    </service>
  </contract>
</serviceActivate>
```

### Sample URL Request

The following URL illustrates how to invoke the same `serviceActivate` operation described above:

```
https://otwsqa.tivo.com/serviceActivate?type=serviceActivate
&contract.0.device.0.hardwareSerialNumber=746321323213214
&contract.0.device.0.siteId=36
&contract.0.device.0.msoServiceId=1
&contract.0.service.0.sku=S00055
&contract.0.customer.0.firstName=Jaya
&contract.0.customer.0.lastName=Marada
&contract.0.customer.0.address.0.street1=2160%20gold%20st
&contract.0.customer.0.address.0.city=Alviso
&contract.0.customer.0.address.0.state=CA
&contract.0.customer.0.address.0.postalCode=95002
&contract.0.customer.0.address.0.country=US
&contract.0.customer.0.partnerCustomerId=234234
&contract.0.customer.0.phone=4085199468
&contract.0.customer.0.email=jmarada@example.com
```

## 6.2.2 Output Element

The `serviceActivate` operation returns either a `serviceActivateResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The `serviceActivate` operation returns a `serviceActivateResponse` element. If the operation triggers an error, the response will include an `error` element; if the operation succeeds, the `error` element is not included.

The following listing shows a `serviceActivateResponse` for a successful `serviceActivate` operation.

```
<serviceActivateResponse>
  <contract>
    <customer>
      <id>0012345678</id>
      <levelOfDetail>high</levelOfDetail>
      <type>customer</type>
    </customer>
    <id>987654321</id>
    <levelOfDetail>high</levelOfDetail>
    <type>contract</type>
  </contract>
  <status>success</status>
  <requestId>6f5fa0fc-9802-44e0aece455c4d4b9a90</requestId>
</serviceActivateResponse>
```

## 6.2.3 Tips and Tricks

To send multiple activations for a new customer:

1. Send a separate request with the provided `partnerCustomerId` to create the initial customer record.
2. Wait for a success response.
3. Send additional activation requests using the same `partnerCustomerId`. This ensures that only one TiVo customer record is associated with the provided `partnerCustomerId`.

# 6.3 serviceCancel Operation

Use the `serviceCancel` operation to cancel TiVo service to a specific device.

**Note:** Canceling service does not delete a customer record. A customer record is never deleted, even if there are no active devices for that customer.

**Note:** To cancel multiple services for a customer, see *serviceCancelByCustomerId Operation* on page 48.

## 6.3.1 Input Parameter

The `serviceCancel` operation requires a `serviceCancelFields` element as an input parameter. The `serviceCancelFields` element wraps a `contract` element that describes the

services being canceled, and a `cancellationCode`, provided to the partner by TiVo. For more information about `serviceCancelFields`, see *serviceCancel Field Group* on page 136. For more information about the `contract` element, see *contract* on page 103.

### Sample XML Input

The following listing shows sample XML input for a `serviceCancel` operation.

```
<serviceCancel>
  <cancellationCode>adba</cancellationCode>
  <contract>
    <customer>
      <partnerCustomerId>234234</partnerCustomerId>
    </customer>
    <device>
      <hardwareSerialNumber>746321323213214</hardwareSerialNumber>
    </device>
  </contract>
</serviceCancel>
```

### Sample URL Request

The following URL invokes a `serviceCancel` operation.

```
https://otwsqa.tivo.com/serviceCancel?type=serviceCancel
&contract.0.device.0.hardwareSerialNumber=746321323213214
&contract.0.customer.0.partnerCustomerId=234234
&cancellationCode=adba
```

## 6.3.2 Output Element

The `serviceCancel` operation returns either a `serviceCancelResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The following listing illustrates a `serviceCancelResponse` for a successful `serviceCancel` operation.

```
<serviceCancelResponse>
  <requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
  <status>success</status>
</serviceCancelResponse>
```

# 6.4 serviceCancelByCustomerId Operation

Cancels all active contracts associated with a specific customer account.

**Note:** Canceling all service does not delete a customer record. A customer record is never deleted, even if there are no active devices for that customer.

## 6.4.1 Input Parameter

The `serviceCancelByCustomerId` operation requires a `serviceCancelByCustsomerIdFields` element, which consists of two fields: `cancellationCode` (provided to a partner by TiVo), and either a `partnerCustomerId` or a `tivoCustomerID`.

### Sample XML Input

The following listing shows sample XML input for a `serviceCancelByCustomerId` operation using a `partnerCustomerId`:

```
<serviceCancelByCustomerId>
  <cancellationCode>MS</cancellationCode>
  <partnerCustomerId>234234</partnerCustomerId>
</serviceCancelByCustomerId>
```

### Sample URL Request

The following URL invokes the `serviceCancelByCustomerId` operation.

```
https://otwsqa.tivo.com/serviceCancelByCustomerId?
type=serviceCancelByCustomerId&cancellationCode=MS&
partnerCustomerId=234234
```

## 6.4.2 Output Element

The `serviceCancelbyCustomerId` operation returns either a `serviceCancelByCustomerIdResponseList` element or an `error` element. On success, the `CancelByCustomerResponseList` identifies each customer device for which service is canceled. Only if the operation triggers an error does the response consist of an `error` element.

For more information about `serviceCancelByCustomerIdResponseList`, see *serviceCancelByCustomerIdResponseList* on page 138.

### Sample XML Response

The following listing shows a `serviceCancelByCustomerIdResponseList` returned by a successful `serviceCancelByCustomerId` operation.

```
<serviceCancelByCustomerIdResponseList>
  <serviceCancelByCustomerIdResponse>
    <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
    </device>
    <status>success</status>
  </serviceCancelByCustomerIdResponse>
```

```
  <serviceCancelByCustomerIdResponse>
    <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
    </device>
    <status>success</status>
  </serviceCancelByCustomerIdResponse>
  <requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
</serviceCancelByCustomerIdResponseList>
```

# 6.5 serviceReset Operation

The `serviceReset` operation resets the service-side data for a specified device. A partner uses this operation in situations where a customer cancels an account and returns a device. The partner can call this operation to reinitialize the device before sending it to a different customer.

**Note:** Resetting service does not delete the customer record originally associated with a device. A customer record is never deleted, even if there are no active devices for that customer.

After performing a `serviceReset` operation, a partner still needs to:

1. Perform a manual procedure to clear and delete any existing device data.
2. Repeat warehouse staging before deploying the device to a new customer.

After these steps, the device enters a "Grace Period" service state for which countdown begins the first time the device is reconnected to the TiVo service.

## 6.5.1 Input Parameter

The `serviceReset` operation requires a `serviceResetFieldGroup` element, which consists of a device element identifying the device to reset. For more information about the `serviceResetFieldGroup`, see *serviceReset Field Group* on page 144.

### Sample XML Input

The following listing shows sample XML input for a `serviceReset` operation.

```
<serviceReset>
  <device>
    <hardwareSerialNumber>746321323213214</hardwareSerialNumber>
  </device>
</serviceReset>
```

### Sample URL Input

The following URL invokes the `serviceReset` operation.

```
https://otwsqa.tivo.com/serviceReset?type=serviceReset
&device.0.hardwareSerialNumber=746321323213214
```

## 6.5.2 Output Element

The `serviceReset` operation returns either a `serviceResetResponse` element or an error element. On success, the `serviceResetResponse` identifies the `requestID` and `status` of the operation. Only if the operation triggers an error does the response consist of an `error` element.If the operation triggers an error, the response will include an `error` element; if the operation succeeds, the `error` element is not included.

### Sample XML Response

The following listing shows a `serviceResetResponse` for a successful `serviceReset` operation.

```
<serviceResetResponse>
  <requestId>f8616c93-41ba-492a-98e8-767dc9350390</requestId>
  <status>success</status>
</serviceResetResponse>
```

# Device Service

7

The Device service provides operations that retrieve configuration and call history information for the device. These operations are grouped into following categories:



The data structures used by these operations are defined in *XML Objects* on page 89.

## 7.1 Operations for Technical Support and Troubleshooting

There are four operations for technical support and troubleshooting, all of which involve querying a device for information:

| Operation | Description | Input | Output useful for troubleshooting |
|---|---|---|---|
| callInfoSearch | Returns the call information for this device. | tivoSerialNumber<br>OR<br>hardwareSerialNumber<br>noOfDays | callInfoSearchResponseList<br>callInformation field group |
| configInfoSearch | Returns the configuration information about this device. | tivoSerialNumber<br>OR<br>hardwareSerialNumber | configInfoSearchResponse<br>configInfo field group |
| deviceInfoSearch | Returns the information about this device. | tivoSerialNumber<br>OR<br>hardwareSerialNumber | deviceInfoSearchResponse<br>device field group<br>drive field group<br>configInfo element |
| serviceGroupFetch | Searches for the custom distribution groups that are assigned to a specific device.<br>NOTE: This operation is not available by default to all partners. | device | serviceGroupFetchResponse |

# 7.2 callInfoSearch Operation

Returns configuration information captured from a specified device each time the device called the TiVo service during an optionally specified number of days.

## 7.2.1 Input Parameters

The `callInfoSearch` operation takes two parameters, one required, and one optional.

Required is a parameter that identifies the device from which to retrieve configuration information, either a `tivoSerialNumber`, or a `hardwareSerialNumber`.

Optional is the `noOfDays` parameter that specifies the number of days (24-hour time periods) for which to return available data. For example, when `noOfDays` is 2, this is equivalent to 48 hours. If you omit `noOfDays`, `callInfoSearch` returns data for the most recent call made by the device to the TiVo service. For more information about returned information, see *Output Element* on page 54.

### Sample XML Input

The following listings show two sample XML input statement for `callInfoSearch` operations. The first example demonstrates how to provided a `tivoSerialNumber` to identify the device for which to return information.

```
<callInfoSearch>
  <noOfDays>1</noOfDays>
  <tivoSerialNumber>746000190230D38</tivoSerialNumber>
</callInfoSearch>
```

The next example uses a `hardwareSerialNumber` to specify the device:

```
<callInfoSearch>
  <hardwareSerialNumber>746000190230D38</hardwareSerialNumber>
  <noOfDays>1</noOfDays>
</callInfoSearch>
```

### Sample URL Request

The following URL examples illustrate how to invoke the `callInfoSearch` operation using a `tivoSerialNumber`:

```
https://otwsqa.tivo.com/callInfoSearch?type=callInfoSearch
&tivoSerialNumber=746000190230D38&noOfDays=1
```

The following URL examples illustrate how to invoke the `callInfoSearch` operation using a `hardwareSerialNumber`:

```
https://otwsqa.tivo.com/callInfoSearch?type=callInfoSearch
&hardwareSerialNumber=746000190230D38&noOfDays=1
```

## 7.2.2 Output Element

The `callInfoSearch` operation returns either a `callInfoSearchResponseList` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

On success, the `callInfoSearchResponse` element always contains at least one structure defining the most recent call information. The actual number of call information structures returned depends on the number of 24-hour time periods specified in the `noOfDays` input parameter and the number of calls the device made to the TiVo services in that specified period of time. For more information about `callInfoSearchResponse`, see *callInfoSearchResponseList* on page 93.

### Sample XML Response

The following listing shows a `callInfoSearchResponseList` for a successful `callInfoSearch` operation.

```
<callInfoSearchResponseList>
<callInfo>
    <ani>--</ani>
    <areaCodeSelected><empty></areaCodeSelected>
    <callTime>11-12-2010 10:24:01</callTime>
    <configInfo>
      <activeUIMode>No Info</activeUIMode>
      <audioSource>Main</audioSource>
      <broadband>false</broadband>
      <cds>Y</cds>
      <dialup>
        <callWaitingPrefix><empty></callWaitingPrefix>
        <dialPrefix><empty></dialPrefix>
        <dialToneCheck>1</dialToneCheck>
        <levelOfDetail>high</levelOfDetail>
        <phoneAvailabilityDetection>No</phoneAvailabilityDetection>
        <pulseDial>Tone</pulseDial>
        <type>dialup</type>
      </dialup>
      <drive>
        <driveModel />
        <driveSize />
        <firmware />
        <married>NO DRIVE REPORTED</married>
        <serialNumber />
        <supported>N/A</supported>
        <type>drive</type>
      </drive>
      <drive1SN><empty></drive1SN>
      <drive2SN><empty></drive2SN>
      <dualTuner>2 (2 tuners)</dualTuner>
      <enableTiVoSuggestions>No</enableTiVoSuggestions>
```

```
      <esata>Y</esata>
      <forceBackhaul>true</forceBackhaul>
      <hardwareSerialNumber>C82860364XBHCGQZ</hardwareSerialNumber>
      <hme>Y</hme>
      <httpAccess>Y</httpAccess>
      <inputSource>
        <headend>0003041</headend>
        <setTopBrand><empty></setTopBrand>
        <type>inputSource</type>
      </inputSource>
      <irDBVersion>-1</irDBVersion>
      <levelOfDetail>high</levelOfDetail>
      <lowerCard />
      <mrv>Y</mrv>
      <musicPhoto>Y</musicPhoto>
      <network>
        <dynamicIPAddress>No Info</dynamicIPAddress>
        <levelOfDetail>high</levelOfDetail>
        <macAddress>0000DEADBEEF</macAddress>
        <networkAdaptor>Debug interface</networkAdaptor>
        <peerToPeerNetwork>No Info</peerToPeerNetwork>
        <physicalConnectionType>No Info</physicalConnectionType>
        <type>network</type>
        <wirelessSignalStrength>-1</wirelessSignalStrength>
      </network>
      <pgdToDate>11-23-2010</pgdToDate>
      <recordQuality>BEST</recordQuality>
      <serviceState>GOOD</serviceState>
      <siteId>98</siteId>
      <swUpgradeAuthorized>false</swUpgradeAuthorized>
      <swVersion>15.1PR.N4-01-3-C00</swVersion>
      <tco>N</tco>
      <terminationCause>Recorder</terminationCause>
      <timeZone>GMT+00:00</timeZone>
      <tivoSerialNumber>CF0010E302ACCBF</tivoSerialNumber>
      <ttcb>Y</ttcb>
      <ttg>Y</ttg>
      <tuningAdapterManufacturer>No info</
tuningAdapterManufacturer>
      <type>configInfo</type>
      <upperCard />
    </configInfo>
    <dialInNumberSelected><empty></dialInNumberSelected>
    <levelOfDetail>high</levelOfDetail>
    <type>callInfo</type>
    <zipcode>GB1-1</zipcode>
  </callInfo>
  <requestId>2325ed8b-05ba-4ecb-98f9-6f320b284f44</requestId>
```

```
  <status>success</status>
</callInfoSearchResponseList>
```

# 7.3 configInfoSearch Operation

Returns device configuration information captured from a specified device when the device last connected to the TiVo Service.

## 7.3.1 Input Parameter

The `configInfoSearch` operation requires a single parameter that identifies the device from which to retrieve configuration information, either a `tivoSerialNumber`, or a `hardwareSerialNumber`.

### Sample XML Input

The following listings show two sample XML input statement for `configInfoSearch` operations. The first example demonstrates how to provided a `tivoSerialNumber` to identify the device for which to return information.

```
<configInfoSearch>
  <tivoSerialNumber>746000190230D38</tivoSerialNumber>
</configInfoSearch>
```

The next example uses a `hardwareSerialNumber` to specify the device:

```
<configInfoSearch>
  <hardwareSerialNumber>746000190230D38</hardwareSerialNumber>
</configInfoSearch>
```

### Sample URL Request

The following URL illustrates how to invoke version 9 of the `configInfoSearch` operation using a `tivoSerialNumber`:

```
https://otwsqa.tivo.com/itmind/mindlite9?type=configInfoSearch
&tivoSerialNumber=746000190230D38
```

## 7.3.2 Output Element

The `configInfoSearch` operation returns either a `configInfoSearchResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

Some of the data returned in `configInfoSearchResponseList` includes the device's hard drive serial numbers, whether the user has enabled TiVo Suggestions, whether the device is configured for the Standard Definition or High Definition TiVo user interface, and whether the user has connected an external hard drive. For more information about `configInfoSearchResponse` see *configInfoSearchResponse* on page 98.

## Sample XML Response

The following listing shows an example of a `configInfoSearchResponse` for a successful `configInfoSearch` operation using version 9 web services. New return elements in version 9 are highlighted in bold.

See the next section for an example response from previous web service versions.

```
<configInfoSearchResponse>
  <configInfo>
    <activeUIMode>No Info</activeUIMode>
    <audioSource>Main</audioSource>
    <broadband>false</broadband>
    <cds>Y</cds>
    <controllerId />
    <dialup>
      <callWaitingPrefix><empty></callWaitingPrefix>
      <dialPrefix><empty></dialPrefix>
      <dialToneCheck>1</dialToneCheck>
      <levelOfDetail>high</levelOfDetail>
      <phoneAvailabilityDetection>No</phoneAvailabilityDetection>
      <pulseDial>Tone</pulseDial>
      <type>dialup</type>
    </dialup>
    <drive>
      <driveModel />
      <driveSize />
      <firmware />
      <married>NO DRIVE REPORTED</married>
      <serialNumber />
      <supported>N/A</supported>
      <type>drive</type>
    </drive>
    <drive1SN><empty></drive1SN>
    <drive2SN><empty></drive2SN>
    <dualTuner>2 (2 tuners)</dualTuner>
    <enableTiVoSuggestions>No</enableTiVoSuggestions>
    <esata>Y</esata>
    <forceBackhaul>true</forceBackhaul>
    <hardwareSerialNumber>C82860364XBHCGQZ</hardwareSerialNumber>
    <hme>Y</hme>
    <httpAccess>Y</httpAccess>
    <inputSource>
      <headend>0003041</headend>
      <headendName />
      <setTopBrand><empty></setTopBrand>
      <type>inputSource</type>
    </inputSource>
    <irDBVersion>-1</irDBVersion>
```

```
    <levelOfDetail>high</levelOfDetail>
    <mrv>Y</mrv>
    <musicPhoto>Y</musicPhoto>
    <network>
      <dynamicIPAddress>No Info</dynamicIPAddress>
      <levelOfDetail>high</levelOfDetail>
      <macAddress>0000DEADBEEF</macAddress>
      <networkAdaptor>Debug interface</networkAdaptor>
      <peerToPeerNetwork>No Info</peerToPeerNetwork>
      <physicalConnectionType>No Info</physicalConnectionType>
      <type>network</type>
      <wirelessSignalStrength>-1</wirelessSignalStrength>
    </network>
    <pgdToDate>11-23-2010</pgdToDate>
    <plantId />
    <recordQuality>BEST</recordQuality>
    <serviceState>GOOD</serviceState>
    <siteId>98</siteId>
    <swUpgradeAuthorized>false</swUpgradeAuthorized>
    <swVersion>15.1PR.N4-01-3-C00</swVersion>
    <tco>N</tco>
    <terminationCause>Recorder</terminationCause>
    <timeZone>GMT+00:00</timeZone>
    <tivoSerialNumber>CF0010E302ACCBF</tivoSerialNumber>
    <ttcb>Y</ttcb>
    <ttg>Y</ttg>
    <tuningAdapterManufacturer>No info</tuningAdapterManufacturer>
    <type>configInfo</type>
  </configInfo>
  <requestId>fdeac93a-278f-4310-ac0c-69913edfe0ef</requestId>
  <status>success</status>
</configInfoSearchResponse>
```

# 7.4 deviceInfoSearch Operation

Returns information captured from the device when it last called the TiVo Service. The information includes the contract details, the customer who currently owns the device, and the device type.

## 7.4.1 Input Parameter

The `deviceInfoSearch` operation requires a single input element that identifies the device from which to retrieve information, either a `tivoSerialNumber`, or a `hardwareSerialNumber`.

### Sample XML Input

The following listings show two sample XML input statement for `deviceInfoSearch` operations. The first example demonstrates how to provided a `tivoSerialNumber` to identify the device for which to return information.

```
<deviceInfoSearch>
  <tivoSerialNumber>746000190230D38</tivoSerialNumber>
</deviceInfoSearch>
```

The next example uses a `hardwareSerialNumber` to specify the device:

```
<deviceInfoSearch>
  <hardwareSerialNumber>746000190230D38</hardwareSerialNumber>
</deviceInfoSearch>
```

### Sample URL Request

The following URL examples illustrate how to invoke version 9 of the deviceInfoSearch operation using a tivoSerialNumber:

```
https://otwsqa.tivo.com/itmind/mind9
?type=deviceInfoSearch&tivoSerialNumber=746000190230D38
```

## 7.4.2 Output Element

The `deviceInfoSearch` operation returns either a `deviceInfoSearchResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

Some of the data returned in `deviceInfoSearchResponse` includes the device's host ID, customer contract, and device type. For more information about `deviceInfoSearchResponse` see *deviceInfoSearchResponse* on page 113.

### Sample XML Response

The following listing shows an example of a `deviceInfoSearchResponse` for a successful `deviceInfoSearch` operation with version 9 of the web service. Fields in **bold** are new to version 9 and are only returned only when invoking the versioned URL starting with `https://host/itmind/mind9`.

```
<deviceInfoSearchResponse>
  <deviceInfo>
    <configInfo>
      <hostId1>0350122423147</hostId1>
      <hostId2>0350122423147</hostId2>
      <levelOfDetail>high</levelOfDetail>
      <type>configInfo</type>
      <tier />
    </configInfo>
    <contract>
      <customer>
```

```
      <address>
        <city>SUNNYVALE</city>
        <country>US</country>
        <levelOfDetail>high</levelOfDetail>
        <postalCode>94086</postalCode>
        <state>CA</state>
        <street1>2190 Main Street</street1>
        <street2 />
        <type>address</type>
      </address>
      <email>tivoacct001@testing123.com</email>
      <firstName>Firstname1000</firstName>
      <id>0012111312</id>
      <lastName>Lastname1000</lastName>
      <levelOfDetail>high</levelOfDetail>
      <optStatus>optNeutral</optStatus>
      <partnerCustomerId>808730001000000</partnerCustomerId>
      <partnerId>0000003755</partnerId>
      <phone>4085551000</phone>
      <type>customer</type>
    </customer>
    <device>
      <levelOfDetail>high</levelOfDetail>
      <partnerId>0000003755</partnerId>
      <subPartnerId />
      <tivoSerialNumber>746000190675BAE</tivoSerialNumber>
      <type>device</type>
    </device>
    <endDate>2011-06-29</endDate>
    <id>0107544728</id>
    <levelOfDetail>high</levelOfDetail>
    <service>
      <description>MSO Monthly Service</description>
      <levelOfDetail>high</levelOfDetail>
      <price>0.0</price>
      <serviceType>service</serviceType>
      <sku>S00055</sku>
      <type>service</type>
    </service>
    <startDate>2011-06-28</startDate>
    <status>Cancelled</status>
    <type>contract</type>
  </contract>
  <deviceType>MSO Premiere 320hr (R74639)</deviceType>
  <hideAdult>Y</hideAdult>
  <levelOfDetail>high</levelOfDetail>
  <mrs>Y</mrs>
  <msoServiceId />
```

```
    <overrideAutoDiscovery>Y</overrideAutoDiscovery>
    <partnerId>0000003755</partnerId>
    <ppv>Y</ppv>
    <purchasePin>Y</purchasePin>
    <sdv>Y</sdv>
    <siteId>03</siteId>
    <subPartnerId>0000000000</subPartnerId>
    <tivoSerialNumber>746000190675BAE</tivoSerialNumber>
    <tuneToApp>Y</tuneToApp>
    <type>device</type>
    <vod>Y</vod>
    <vpn>Y</vpn>
  </deviceInfo>
  <manufacturingInfo>
    <ccDate>Nov 22 2011 10:22:52:000PM</ccDate>
    <ccId>2224448912748</ccId>
    <ccManufacturer>MOTO</ccManufacturer>
    <ccSerialNumber>PKTVOAAMT</ccSerialNumber>
    <ccShipTo>18372</ccShipTo>
    <ccUnitAddress>2224448912999</ccUnitAddress>
    <dataId>2220119954111</dataId>
    <hostId>0222119954302</hostId>
    <levelOfDetail>high</levelOfDetail>
    <macAddress>22:22:D9:4C:03:8E</macAddress>
    <test5Date>Jan 22 2011 10:22:19:000PM</test5Date>
    <tivoSerialNumber>746567843998655</tivoSerialNumber>
    <type>manufacturing</type>
  </manufacturingInfo>
  <requestId>3a5c1948-f1e9-4719-be63-3385ee010806</requestId>
  <status>success</status>
</deviceInfoSearchResponse>
```

# 7.5 serviceGroupFetch Operation

Searches for the custom distribution groups that are assigned to a specific device, such as dial code group.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

## 7.5.1 Input Parameter

The `serviceGroupFetch` operation requires a single parameter, `device`, that identifies the device from which to retrieve information.

### Sample XML Input

The following listing shows a sample XML input for a `serviceGroupFetch` operation.

```
<serviceGroupFetch>
  <device>
    <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
<serviceGroupFetch>
```

### Sample URL Request

The following URL invokes the `serviceGroupFetch` operation.

```
https://otwsqa.tivo.com/serviceGroupFetch?type=serviceGroupFetch
&device.0.hardwareSerialNumber=648001180200100
```

## 7.5.2 Output Element

The `serviceGroupFetch` operation returns either a `serviceGroupFetchResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The following listing shows a `serviceGroupFetchResponse` for a successful `serviceGroupFetch` operation.

```
<serviceGroupFetchResponse>
  <group>
    <groupName>combo</groupName>
    <groupPrefix>DG</groupPrefix>
  </group>
  <group>
    <groupName>combo</groupName>
    <groupPrefix>DG</groupPrefix>
  </group>
  <group>
    <groupName>combo</groupName>
    <groupPrefix>DG</groupPrefix>
  </group>
  <group>
    <groupName>combo</groupName>
    <groupPrefix>DG</groupPrefix>
  </group>
  <requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
  <status>success</status>
</serviceGroupFetchResponse>
```

### Error Response Example

The following listing shows an `error` for a `serviceGroupFetch` operation that triggered an error.

```
<error>
  <code>1011</code>
  <text>Hardware Serial Number not found</text>
  <requestId>1234567890-abc-1234567890</requestId>
</error>
```

## 7.6 Operations for Configuring or Setting a Feature

There are two operations for configuring or setting device features:

| Operation Name | Description | Input | Output |
|---|---|---|---|
| deviceInfoStore | Store the name and settings of the device. | device | deviceInfoStoreResponse |
| dialCodeUpdate | Assign a device to a specific Dialing Code (DC) group based on the environment (env) parameter. | tivoSerialNumber<br>hardwareSerialNumber<br>env | dialCodeUpdateResponse |

### 7.6.1 deviceInfoStore Operation

Stores configuration data and the friendly name for a specified device.

#### 7.6.1.1 Input Parameter

The `deviceInfoStore` operation requires a single element, `device`, that identifies the device for which to set configuration information. The `device` element provides settings that determine access to various TiVo services, provides a user-friendly device name, and the TiVo serial number (TSN) for the device. For more information about `device`, see *device* on page 110.

### Sample XML Input

The following listings show sample XML input for a `deviceInfoStore` operation.

```
<deviceInfoStore>
  <device>
    <configInfo>
      <cds>Y</cds>
      <esata>Y</esata>
      <hme>N</hme>
      <httpAccess>Y</httpAccess>
      <mrv>Y</mrv>
      <musicPhoto>N</musicPhoto>
      <tco>Y</tco>
```

```
      <ttcb>Y</ttcb>
      <ttg>Y</ttg>
    </configInfo>
    <hideAdult>Y</hideAdult>
    <mrs>Y</mrs>
    <name>Jaya DVR</name>
    <overrideAutoDiscovery>Y</overrideAutoDiscovery>
    <ppv>Y</ppv>
    <purchasePin>Y</purchasePin>
    <sdv>Y</sdv>
    <siteId>36</siteId>
    <tier>non-PVR</tier>
    <tivoSerialNumber>746000190230D38</tivoSerialNumber>
    <vod>Y</vod>
    <vpn>Y</vpn>
  </device>
</deviceInfoStore>
```

## Sample URL Input

The following URL invokes the `deviceInfoStore` operation.

```
https://otwsqa.tivo.com/
deviceInfoStore?type=deviceInfoStore&device.0.tivoSerialNumber=B420
001802583EE
&device.0.configInfo.0.mrv=y
&device.0.configInfo.0.musicPhoto=n
&device.0.configInfo.0.ttg=Y
&device.0.configInfo.0.ttcb=y
&device.0.configInfo.0.hme=N
&device.0.configInfo.0.httpAccess=y
&device.0.configInfo.0.tco=Y
&device.0.configInfo.0.cds=n
&device.0.configInfo.0.esata=y
&device.0.name=Jaya DVR
&device.0.siteId=36
&device.0.mrs=y
&device.0.hideAdult=y
&device.0.ppv=y
&device.0.purchasePin=y
&device.0.vod=y
&device.0.sdv=y
&device.0.vpn=y
&device.0.overrideAutoDiscovery=y
&device.0.tier=non-PVR
```

#### 7.6.1.2 Output Element

The `deviceInfoStore` operation returns either a `deviceInfoStoreResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

#### Sample XML Response

The following listing shows a `deviceInfoStoreResponse` for a successful `deviceInfoStore` operation.

```
<deviceInfoStoreResponse>
  <requestId>xxxxxx</requestId>
  <status>success</status>
</deviceInfoStoreResponse>
```

### 7.6.2 dialCodeUpdate Operation

Specifies the TiVo Service center to which a device connects: production or staging.

**Note:** This operation is only applicable for partners who operate their own service centers and must move boxes between test and production environments. This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

#### 7.6.2.1 Input Parameters

The `dialCodeUpdate` operation requires two parameters: either the `hardwareSerialNumber` or `tivoSerialNumber` that identifies the device; and `env`, which specifies the TiVo Service center to which the device should connect.

#### Sample XML Input

The following listings show sample XML input for a `dialCodeUpdate` operation. Example 1 specifies that a device connects to the production environment. Example 2 specifies that a device connects to the staging (test) environment.

**Example 1**

```
<dialCodeUpdate>
  <env>production</env>
  <hardwareSerialNumber>C80860217XBFCDVQ</hardwareSerialNumber>
</dialCodeUpdate>
```

**Example 2**

```
<dialCodeUpdate>
  <env>staging</env>
  <hardwareSerialNumber>C80860217XBFCDVQ</hardwareSerialNumber>
</dialCodeUpdate>
```

### Sample URL Request

The following URLs invoke the `dialCodeUpdate` operation using different parameters.

#### Example 1

```
https://otwsqa.tivo.com/dialCodeUpdate?
type=dialCodeUpdate&hardwareSerialNumber=C80860217XBFCDVQ
&env=production
```

#### Example 2

```
https://otwsqa.tivo.com/dialCodeUpdate?
type=dialCodeUpdate&hardwareSerialNumber=C80860217XBFCDVQ&env=staging
```

### 7.6.2.2 Output Element

The `dialCodeUpdate` operation returns either a `dialCodeUpdateResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The following listing shows a `dialCodeUpdateResponse` for a successful `dialCodeUpdate` operation.

```
<dialCodeUpdateResponse>
  <requestId>52617b6b-154d-49e3-9b97-e7e870ea37b2</requestId>
  <status>success</status>
</dialCodeUpdateResponse>
```

### Error Response Example

The following listing shows an `error` for a `dialCodeUpdate` operation that triggers an error.

```
<error>
  <code>1011</code>
  <text>Hardware Serial Number not found</text>
  <requestId>1234567890-abc-1234567890</requestId>
</error>
```

## 7.7 Operations for Managing Service Groups

There are two operations for managing service groups.

| Operation Name | Description | Input | Output |
|---|---|---|---|
| serviceGroupRemove | Removes one or more devices from an existing custom distribution group. | device<br>group | serviceGroupRemoveResponseList |

| Operation Name | Description | Input | Output |
|---|---|---|---|
| serviceGroupStore | Adds one or more devices to an existing custom distribution group. | device<br>group | serviceGroupStoreResponseList |

## 7.7.1 serviceGroupRemove Operation

Removes one or more devices from an existing custom distribution group.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

### 7.7.1.1 Input Parameters

The `serviceGroupRemove` operation requires at least two parameters: a `device` element specifying a device to remove; and a `group` element that lists the custom distribution group from which to remove the device. The `device` element wraps either a `hardwareSerialNumber` or `tivoSerialNumber` parameter that uniquely identifies that device to remove.

**Note:** Attempting to remove a device from a default group results in an error.

To remove multiple devices from a custom distribution group, a `serviceGroupRemove` operation can specify separate `device` elements for each device to remove from a group.

#### Sample XML Input

The following listing show sample XML input for a `serviceGroupRemove` operation to remove three devices a custom distribution group:

```
<serviceGroupRemove>
  <device>
    <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <device>
    <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <device>
      <tivoSerialNumber>648001180200100</tivoSerialNumber>
  </device>
  <group>
    <groupName>combo</groupName>
    <groupPrefix>DG</groupPrefix>
  </group>
</serviceGroupRemove>
```

### Sample URL Request

The following URL invokes the `serviceGroupRemove` operation to remove two devices from a custom distribution group:

```
https://otwsqa.tivo.com/serviceGroupRemove?type=serviceGroupRemove
&group.0.groupName=combo
&group.0.groupPrefix=DG
&device.0.hardwareSerialNumber=648001180200100
&device.1.hardwareSerialNumber=648001180200100
```

## 7.7.1.2 Output Element

For `serviceGroupRemove` operations that attempt to remove a single device, the operations return either a `serviceGroupRemoveResponseList` element, or an `error` element. For operations that specify multiple devices to remove where not all removals are successful, the `serviceGroupRemove` operations return both a `serviceGroupRemoveResponseList` and an `error` element.

Only if the operation triggers an error does the response contain an `error` element.

### Sample XML Response

The following listing shows a `serviceGroupRemoveResponseList` for a successful `serviceGroupRemove` operation.

#### Example 1 - Success

```
<serviceGroupRemoveResponseList>
  <serviceGroupRemoveResponse>
    <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
    </device>
    <status>success</status>
  </serviceGroupRemoveResponse>
  <serviceGroupRemoveResponse>
    <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
    </device>
    <status>success</status>
  </serviceGroupRemoveResponse>
  <requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
</serviceGroupRemoveResponseList>
```

#### Example 2 - Partial Success

```
<serviceGroupRemoveResponseList>
  <serviceGroupRemoveResponse>
    <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
    </device>
  <status>success</status>
```

```
    </serviceGroupRemoveResponse>
    <serviceGroupRemoveResponse>
  <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <status>success</status>
  </serviceGroupRemoveResponse>
    <serviceGroupRemoveResponse>
  <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <error>
    <code>1011</code>
    <text>Hardware Serial Number not found</text>
  </error>
  <status>error</status>
  </serviceGroupRemoveResponse>
  <requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
</serviceGroupRemoveResponseList>
```

### Error Response Example

The following listing shows an `error` for a `serviceGroupRemove` operation that triggers an error.

```
<error>
  <code>1011</code>
  <text>Hardware Serial Number not found</text>
  <requestId>1234567890-abc-1234567890</requestId>
</error>
```

## 7.7.2 serviceGroupStore Operation

Adds one or more devices to an existing custom distribution group.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

### 7.7.2.1 Input Parameters

The `serviceGroupStore` operation requires at least two parameters: a `device` element specifying a device to add to a custom distribution group; and a `group` element that lists the custom distribution group to which to add the device. The `device` element wraps either a `hardwareSerialNumber` or `tivoSerialNumber` parameter that uniquely identifies that device to add.

**Note:** Attempting to add a device to a default distribution group rather than a custom distribution group results in an error.

To add multiple devices to a custom distribution group, a `serviceGroupStore` operation can specify separate `device` elements for each device to add to a group.

### Sample XML Input

The following listing show sample XML input for a `serviceGroupStore` operation.

```
<serviceGroupStore>
  <device>
  <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <device>
  <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <device>
    <tivoSerialNumber>648001180200100</tivoSerialNumber>
  </device>
  <group>
    <groupName>combo</groupName>
    <groupPrefix>DG</groupPrefix>
  </group>
</serviceGroupStore>
```

### Sample URL Input

The following URL invoke the `serviceGroupStore` operation using different parameters.

```
https://otwsqa.tivo.com/serviceGroupStore
?type=serviceGroupStore&group.0.groupName=combo
&group.0.groupPrefix=DG
&device.0.hardwareSerialNumber=648001180200100
&device.1.hardwareSerialNumber=648001180200100
```

## 7.7.2.2 Output Element

For `serviceGroupStore` operations that add a single device, the operations return either a `serviceGroupStoreResponseList` element, or an `error` element. For operations that specify multiple devices where not all add operations are successful, the `serviceGroupStore` operations return both a `serviceGroupStoreResponseList` and an `error` element.

Only if the operation triggers an error does the response contain an `error` element.

### Sample XML Response

The following listing shows a `serviceGroupStoreResponseList` for a successful `serviceGroupStore` operation.

#### Example 1 - Success

```
<serviceGroupStoreResponseList>
  <serviceGroupStoreResponse>
```

```
  <device>
    <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <status>success</status>
  </serviceGroupStoreResponse>
  <serviceGroupStoreResponse>
      <device>
      <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <status>success</status>
  </serviceGroupStoreResponse>
  <requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
</serviceGroupStoreResponseList>
```

**Example 2 - Partial Success**

```
<serviceGroupStoreResponseList>
  <serviceGroupStoreResponse>
  <device>
    <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <status>success</status>
  </serviceGroupStoreResponse>
  <serviceGroupStoreResponse>
  <device>
    <hardwareSerialNumber>648001180200100</hardwareSerialNumber>
  </device>
  <error>
    <code>1011</code>
    <text>Hardware Serial Number not found</text>
  </error>
  <status>error</status>
</serviceGroupStoreResponse>
<requestId>d0a2cc63-2334-4b94-a5e9-fad16862482c</requestId>
</serviceGroupStoreResponseList>
```

## Error Response Example:

The following listing shows a `serviceGroupStoreResponse` for a `serviceGroupStore` operation that triggers an error.

```
<error>
  <code>1011</code>
  <text>Hardware Serial Number not found</text>
  <requestId>1234567890-abc-1234567890</requestId>
</error>
```

# 7.8 Operations for Managing On-device Service Messages

There are four operations for managing on-device service messages.

| Operation Name | Description | Input | Output |
|---|---|---|---|
| serviceMessageStore | Creates or updates a Pre-TiVo Central Message (PTCM) or Message Board Message (MBM). | messageId<br>messageType<br>messageText<br>expiryDate<br>from<br>subject | serviceMessageStoreResponse |
| serviceMessageRemove | Makes a Pre-TiVo Central Message (PTCM) or a Message Board Message (MBM) expire. | messageId | serviceMessageRemoveResponse |
| serviceMessageSearch | Search for a specific PTCM or MBM by the message ID. This operation can also be used to search for all PTCMs or MBMs created by a partner. | messageId<br>messageText<br>messageStatus | serviceMessageSearchResponseList |
| deviceMessageSend | Sends a specific PTCM or MBM to a specific device or to a list of devices. | tivoSerialNumber<br>hardwareSerialNumber<br>messageId | deviceMessageResponseList |

## 7.8.1 serviceMessageStore Operation

The `serviceMessageStore` operation creates or updates a Pre-TiVo Central Message (PTCM) or a Message Board Message (MBM).

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

### 7.8.1.1 Input Parameters

A `serviceMessageStore` operation that creates a new message must include two parameters, `messageText`, and `messageType`. A `serviceMessageStore` operation that updates an existing message must specify the `messageId` parameter.

A `from` parameter is a text string that identifies the originator of the message. Space characters in the string must be escaped using`%20` syntax.

A `subject` parameter is a text string that provides a message header. Space characters in the string must be escaped using`%20` syntax.

An optional parameter, `expiryDate`, may be provided when creating or updating a message.

The `messageText` parameter is a text string containing the message. Space characters in the string must be escaped using`%20` syntax. When updating a message, you can change the message text by passing in a different `messageText` parameter.

The `messageType` parameter is one of `ptcm` or `mbm`.

The `messageID` parameter is a unique identifier for the message.

The optional `expiryDate` specifies a date in the future when a message expires. If the expiration date is omitted when you create a new message, then the expiration date is set by default to 1 year from the current date.

## Sample XML Input

The following listing shows XML input for creating a message.

```
<serviceMessageStore>
  <expiryDate>12-31-2012</expiryDate>
  <from>From%20Partner%20Customer%20Support</from>
  <messageText>Testing%20sending%20a%20message</messageText>
  <naturalistically</messageType>
  <subject>This%20is%20a%20test%20message</subject>
</serviceMessageStore>
```

The following listing shows XML input for updating an existing message.

```
<serviceMessageStore>
  <messageId>1001</messageId>
  <messageText>This%20is%20an%20update%20test</messageText>
</serviceMessageStore>
```

## Sample URL Request

The following URL invokes the `serviceMessageStore` operation to create a message.

```
https://otwsqa.tivo.com/
serviceMessageStore?type=serviceMessageStore
&from=From%20Partner%20Customer%20Support
&subject=Testing%20sending%20a%20message
&messageText=This%20is%20a%20test%20message
&messageType=ptcm
&expiryDate=12-31-2012
```

The following URL invokes the `serviceMessageStore` operation to update an existing message.

```
https://otwsqa.tivo.com/
serviceMessageStore?type=serviceMessageStore
&messageId=1001
&messageText=This%20is%20an%20update%20test
&expiryDate=12-31-2012
```

#### 7.8.1.2 Output Element

The `serviceMessageStore` operation returns either a `serviceMessageStoreResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

A successful `serviceMessageStore` operation returns a unique `messageId` for the message. Partners should note this ID since it's needed for subsequent operations to modify or remove messages and to send messages to a device.

#### 7.8.1.3 Sample XML Response

The following listing shows a `serviceMessageStoreResponse` for a `serviceMessageStore` operation that successfully creates a message.

```
<serviceMessageStoreResponse>
  <messageId>1001</messageId>
  <requestId>2e32e921-0a16-4e01-b029-6cd6b049a631</requestId>
  <status>success</status>
</serviceMessageStoreResponse>
```

The following listing shows a `serviceMessageStoreResponse` for a `serviceMessageStore` operation that successfully updates an existing message.

```
<serviceMessageStoreResponse>
  <messageId>1001</messageId>
  <requestId>af6c9c25-a2bc-4d22-892e-1b3068b81c61</requestId>
  <status>success</status>
</serviceMessageStoreResponse>
```

### 7.8.2 serviceMessageRemove Operation

Causes a PTCM or a MBM to expire. Expired messages are not displayed.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

#### 7.8.2.1 Input Parameter

The `serviceMessageRemove` operation requires a single parameter, the `messageID` of the message to remove. This operation sets the expiration date of the message to one day prior to the current date.

#### Sample XML Input

```
<serviceMessageRemove>
  <messageId>1001</messageId>
</serviceMessageRemove>
```

### Sample URL Request

The following URLs invoke the `serviceMessageRemove` operation.

```
https://otwsqa.tivo.com/
serviceMessageRemove?type=serviceMessageRemove&messageId=1001
```

#### 7.8.2.2 Output Element

The `serviceMessageRemove` operation returns either a `serviceMessageRemoveResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

### Sample XML Response

The following listing shows a `serviceMessageRemoveResponse` for a successful `serviceMessageRemove` operation.

```
<serviceMessageRemoveResponse>
  <requestId>2a911c60-b504-4ad5-9075-dc7776921e70</requestId>
  <status>success</status>
</serviceMessageRemoveResponse>
```

## 7.8.3 serviceMessageSearch Operation

Search for a specific PTCM or MBM by the message ID or message text, or search for all PTCMs or MBMs created by a partner that have a specified status.

#### 7.8.3.1 Input Parameters

A single call to the serviceMessageSearch operation takes one of three parameters: `messageId`, `messageText`, or `messageStatus`.

To search for a specific message by ID, pass `messageId`.

To search for a specific message by its text, pass `messageText`.

To search for all partner messages with a specific status, pass `messageStatus`.

### Sample XML Input

The following listing shows XML input to search for a specific message by message ID.

```
<serviceMessageSearch>
  <messageId>1001</messageId>
</serviceMessageSearch>
```

The following listing shows the XML input to search for messages by message status.

```
<serviceMessageSearch>
  <messageStatus>Active</messageStatus>
</serviceMessageSearch>
```

### Sample URL Request

The following URL invokes the `serviceMessageSearch` operation to search for a message with the specified message ID.

```
https://otwsqa.tivo.com/
serviceMessageSearch?type=serviceMessageSearch
&messageId=1001
```

The following URL invokes the `serviceMessageSearch` operation to search for messages with the specified message status.

```
https://otwsqa.tivo.com/
serviceMessageSearch?type=serviceMessageSearch
&messageStatus=Active
```

## 7.8.3.2 Output Element

The `serviceMessageSearch` operation returns either a `serviceMessageSearchResponseList` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

For messages searched by `messageStatus`, the `serviceMessageSearchResponseList` may return more than one `serviceMessage`.

### Sample XML Response

The following listing shows a `serviceMessageSearchResponse` for a successful `serviceMessageSearch` by message ID.

```
<serviceMessageSearchResponse>
  <requestId>60f49afd-3e08-4d8f-874d-0f453ccc07ed</requestId>
  <serviceMessageList>
    <serviceMessage>
      <expiryDate>2012-12-31 00:00:00</expiryDate>
      <levelOfDetail>high</levelOfDetail>
      <messageText>
        From: From Partner Customer Support
        Subject: This is a test message
        ExpirationDate: 15705
        Destination: ptcm
        Body: Testing sending a message
      </messageText>
      <type>serviceMessage</type>
    </serviceMessage>
    <type>serviceMessageList</type>
  </serviceMessageList>
  <status>success</status>
</serviceMessageSearchResponse>
```

The following listing shows a `serviceMessageSearchResponse` for a successful `serviceMessageSearch` by message status.

```
<serviceMessageSearchResponse>
  <requestId>60f49afd-3e08-4d8f-874d-0f453ccc07ed</requestId>
  <serviceMessageList>
    <serviceMessage>
      <expiryDate>2012-12-31 00:00:00</expiryDate>
      <levelOfDetail>high</levelOfDetail>
      <messageText>
        From: From Partner Customer Support
        Subject: This is a test message
        ExpirationDate: 15705
        Destination: ptcm
        Body: Testing sending a message
      </messageText>
      <type>serviceMessage</type>
    </serviceMessage>
    <serviceMessage>
      <expiryDate>2012-12-31 00:00:00</expiryDate>
      <levelOfDetail>high</levelOfDetail>
      <messageText>From: From Customer Support
        Subject: Sending a message
        ExpirationDate: 15705
        Destination: mbm
        Body: Sending a test message
      </messageText>
      <type>serviceMessage</type>
    </serviceMessage>
    <type>serviceMessageList</type>
  </serviceMessageList>
  <status>success</status>
</serviceMessageSearchResponse>
```

## 7.8.4 deviceMessageSend Operation

Sends a specific message (MBM or PTCM) to one or more devices.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

### 7.8.4.1 Input Parameters

The `deviceMessageSend` operation requires a minimum of two parameters: a `messageID` specifying the message to send to one or more devices; and one or more devices, identified either by `tivoSerialNumber` or `hardwareSerialNumber`.

To send a message to multiple devices, separate device IDs by commas.

### Sample XML Input

The following listing shows XML input to send a specific message to a device.

```
<deviceMessageSend>
  <messageId>1001</messageId>
  <tivoSerialNumbers>B420000001E46B2
    B420001902555AE</tivoSerialNumbers>
</deviceMessageSend>
```

### Sample URL Request

The following URL invokes the `deviceMessageSend` operation:

```
https://otwsqa.tivo.com/deviceMessageSend?
type=deviceMessageSend&messageId=1001
&tivoSerialNumbers=B420000001E46B2,B420001902555AE
```

## 7.8.4.2 Output Element

The `deviceMessageSend` operation returns a `deviceMessageSendResponseList` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element. For example, the operations returns an error if an expired message is specified as a parameter in the operation.

For messages sent to more than one device, a `deviceMessageSendResponseList` contains one `deviceMessageSendResponse` for each device. In some cases an operation may result in partial success, where a message sent to one device is successful, but a message sent to another is unsuccessful. In these cases a `deviceMessageSend` sub-element may include an `error` element as in the example below.

### Sample XML Response

The following listing shows a `deviceMessageSendResponseList` for a partially successful `deviceMessageSend` operation.

```
<deviceMessageSendResponseList>
  <deviceMessageSendResponse>
    <device>
      <tivoSerialNumber>B420000001E46B3</tivoSerialNumber>
    </device>
    <status>success</status>
    </deviceMessageSendResponse>
    <deviceMessageSendResponse>
      <device>
        <tivoSerialNumber>B420000001E46B4</tivoSerialNumber>
      </device>
      <error>
        <code>1111</code>
        <text>Serial number not belonging to partner</text>
      </error>
```

```
    </deviceMessageSendResponse>
  <requestId>cfb61c46-8df9-4180-86a0-7a01fd189010</requestId>
</deviceMessageSendResponseList>
```

# Online Service

8

The Online service is used to request a recording of specific broadcast content on a specific device, search for content, and search on recordings.

For recording requests, the partner provides customers a user interface to search for and select upcoming broadcasts of programs, and uses the Online service to schedule recordings of the selected content. This content may be a single episode of a program, or a Season Pass® (typically a series of episodes, for example, *The Good Guys*).

The data structures used by these operations are defined in *XML Objects* on page 89.

## 8.1 Operations

The following table summarizes the Online service operations.

| Operation Name | Description | Input | Output |
|---|---|---|---|
| contentSearch | Retrieves the program details and the similar shows for the specified content. | contentSearch Field Group | contentSearchResponse |
| recordingSearch | Search for recordings that meet specified criteria. | recordingSearch Fields | recordingSearchResponse |
| scheduleRecordingStore | The request is a new scheduleRecording object. The response is the updated scheduleRecording object which includes the assigned schedule RecordingId. | scheduleRecording Fields | scheduleRecordingRespons e |

**Note:** These operations are not available by default to all TiVo partners. If you need to use them, check the IT Release Notes to confirm their availability. If you need to request access to any of these operations, contact your TiVo account manager.

### 8.1.1 contentSearch Operation

Searches for and returns program details for a specific show or series, and optinally returns program details for similar show. This operation is used in tivo.com for the "If you like this…" feature.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

#### 8.1.1.1 Input Parameter

The `contentSearch` operation requires one element, a `contenSearch` field group that describes the content for which to search. For more information about the `contentSearch` field group, see *contentSearch Field Group* on page 101.

### Sample XML Input

The following listings show sample XML input for retrieving a program's details and details about any similar shows.

```
<contentSearch>
  <collectionType>series</collectionType>
  <note>availableCorrelatedCollectionForCollectionId</note>
  <partnerContentId>epgProvider:ct.EP0191774755</partnerContentId>
  <tivoSerialNumber>648001180200100</tivoSerialNumber>
</contentSearch>
```

### Sample URL Request

The following URL invokes the `contentSearch` operation to retrieve a program's details and details about any similar shows.

```
http://otwsqa.tivo.com/contentSearch?type=contentSearch
&tivoSerialNumber=648001180200100
&partnerContentId=epgprovider:ct.123456
&collectionType=series
&note=availableCorrelatedCollectionForCollectionId
```

#### 8.1.1.2 Output Element

The `contentSearch` operation returns a `contentSearchResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

When a `contentSearch` request specifies a `partnerContentId` (as in the *Sample XML Input* on page 81 and *Sample URL Request* on page 81), the response includes a `contentList` data structure that contains a `content` element. If there are related programs, the `content` element contains one `availableCorrelatedCollectionForCollectionId` element for each related content item.

### Sample XML Response

The following listing shows a `contentSearchResponse` for a successful `contentSearch` operation.

```
<contentSearchResponse>
<requestId>xxxxxxxx</requestId>
<status>success</status>
<contentList>
<content>
```

```
<autoOverriddenContentId>tivo:ct.47819941
  </autoOverriddenContentId>
<availableCorrelatedCollectionForCollectionId>
  <collectionId>tivo:cl.55067183</collectionId>
  <collectionType>series</collectionType>
  <episodic>true</episodic>
  <levelOfDetail>low</levelOfDetail>
  <objectIdAndType>344147194561071</objectIdAndType>
<title>Bones</title>
  <type>collection</type>
</availableCorrelatedCollectionForCollectionId>
<availableCorrelatedCollectionForCollectionId>
  <collectionId>tivo:cl.140399202</collectionId>
  <collectionType>series</collectionType>
  <episodic>true</episodic>
  <levelOfDetail>low</levelOfDetail>
  <objectIdAndType>344147194561071</objectIdAndType>
  <title>Bones</title>
  <type>collection</type>
</availableCorrelatedCollectionForCollectionId>
<availableCorrelatedCollectionForCollectionId>
  <collectionId>tivo:cl.55067183</collectionId>
  <collectionType>series</collectionType>
  <episodic>true</episodic>
  <levelOfDetail>low</levelOfDetail>
  <objectIdAndType>344147194561071</objectIdAndType>
  <title>Bones</title>
  <type>collection</type>
</availableCorrelatedCollectionForCollectionId>
<collectionDescription>
  A brilliant and acerbic diagnostician
  leads a team of specialists.
</collectionDescription>
<collectionId>tivo:cl.34115777</collectionId>
<collectionType>series</collectionType>
<contentId>tivo:ct.ts.2646311</contentId>
<contentType>video</contentType>
<description>
  House and his team tackle the case
  of a young man with a penchant for odd behavior,
  and House prepares for his date with Cameron.
</description>
<episodeNum>20</episodeNum>
<episodic>true</episodic>
<imageUrl>/images/zap2it/programs/185044/p185044_ce_h7_aa.jpg
  </imageUrl>
<isEpisode>true</isEpisode>
<levelOfDetail>medium</levelOfDetail>
```

```
  <partnerContentId>180392000-HOU-01-E5520</partnerContentId>
  <partnerId>tivo:pt.2990</partnerId>
  <seasonNumber>1</seasonNumber>
  <subtitle>Love Hurts</subtitle>
  <title>House</title>
  <type>content</type>
</content>
</contentList>
</contentSearchResponse>
```

## 8.1.2 recordingSearch Operation

Returns recording lists for a device since the last time the device synchronized its recordings with the TiVo Service. Recording lists returned include Now Playing and To Do lists.

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

### 8.1.2.1 Input Parameter

The `recordingSearch` operation requires a single `recordingSearch` element that describe the recordings to search for, and specifies the device to search on. For more information about the recordingSearch element, see *recordingSearch* on page 130.

#### Sample XML Input

The following listing shows sample XML input for a `recordingSearch` operation.

```
<recordingSearch>
  <count>1</count>
  <groupBy>collectionId</groupBy>
  <minStartTime>2009-02-04 20:00:00</minStartTime>
  <offset>0</offset>
  <orderBy>subtitle</orderBy>
  <state>complete</state>
  <tivoSerialNumber>648001180200100</tivoSerialNumber>
<recordingSearch>
```

#### Sample URL Request

The following URL invokes a `recordingSearch` operation.

```
http://otwsqa.tivo.com/recordingSearch?type=recordingSearch
&tivoSerialNumber=648001180200100
&state=complete
&count=1
&minStartTime=2009-020420:00:00
&offset=0
```

```
&groupBy=collectionId
&orderBy=subtitle
```

### 8.1.2.2 Output Element

The `recordingSearch` operation returns a `recordingSearchResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element.

A successful `recordingSearchResponse` includes both a TiVo-specific `contentId` element and a partner-specific `partnerContentId` element. For more information about the `recordingSearchResponse` element, see *recordingSearchResponse* on page 131.

### Sample XML Response

The following listing shows a `recordingSearchResponse` for a successful `recordingSearch` operation that does not include the optional `groupBy` parameter.

```
<recordingSearchResponse>
  <requestId>xxxxxxxx</requestId>
  <status>success</status>
  <recordingList>
    <isBottom>false</isBottom>
    <isTop>true</isTop>
    <recording>
      <tivoSerialNumber> 648001180200100</tivoSerialNumber>
      <channel>
        <callSign>KBCW</callSign>
        <channelNumber>12</channelNumber>
        <levelOfDetail>low</levelOfDetail>
        <sourceType>cable</sourceType>
        <stationId>tivo:st.454</stationId>
        <partnerStationId>epgProvider:st.24344</partnerStationId>
        <type>channel</type>
      </channel>
      <collectionId>tivo:cl.15566705</collectionId>
      <contentId>tivo:ct.33082087</contentId>
      <contentType>video</contentType>
      <deletionPolicy>neverDelete</deletionPolicy>
      <duration>1800</duration>
      <episodeNum>4</episodeNum>
      <episodic>true</episodic>
      <expectedDeletion>2038-01-19 00:00:00</expectedDeletion>
      <hdtv>false</hdtv>
      <isEpisode>true</isEpisode>
      <levelOfDetail>low</levelOfDetail>
      <mimeType>video/mpg2</mimeType>
      <partnerCollectionId>epgProvider:cl.SH019177
        </partnerCollectionId>
```

```
      <partnerContentId>epgProvider:ct.EP0191774755
        </partnerContentId>
      <recordingId>tivo:rc.3329</recordingId>
      <requestedEndPadding>1800</requestedEndPadding>
      <requestedStartPadding>600</requestedStartPadding>
      <seasonNumber>2</seasonNumber>
      <startTime>2009-09-23 05:30:00</startTime>
      <state>complete</state>
      <subtitle>Go Get Mommy's Bra</subtitle>
      <title>Two and a Half Men</title>
      <type>recording</type>
    </recording>
  </recordingList>
</recordingSearchResponse>
```

When the request includes the optional `groupBy` parameter, the response is different. The recording elements) are contained within a `recordingGroup` element that is, itself, contained in a `recordingGroupList` element.

```
<recordingSearchResponse>
  <requestId>xxxxxxxx</requestId>
  <status>success</status>
  <recordingGroupList>
  <recordingGroup>
    <count>2</count>
    <example>…..</example>
    <example>…..</example>
  </recordingGroup>
  <recordingGroup>……</recordingGroup>
  </recordingGroupList>
<recordingSearchResponse>
```

**Note:** The content of the `example` element in this sample code has been omitted. The `example` element is identical in structure to the `record` element in the prior example.

## 8.1.3 scheduleRecordingStore Operation

Requests that specific broadcast content be recorded on a specific device. This content may be a single episode of a program, or a Season Pass™ for a series of episodes (for example, *Lost*).

**Note:** This operation is not available by default to all TiVo partners. If you need to use this operation, check the IT Release Notes to confirm its availability. If you need to request access to the operation, contact your TiVo account manager.

### 8.1.3.1 How Remote Scheduling Works

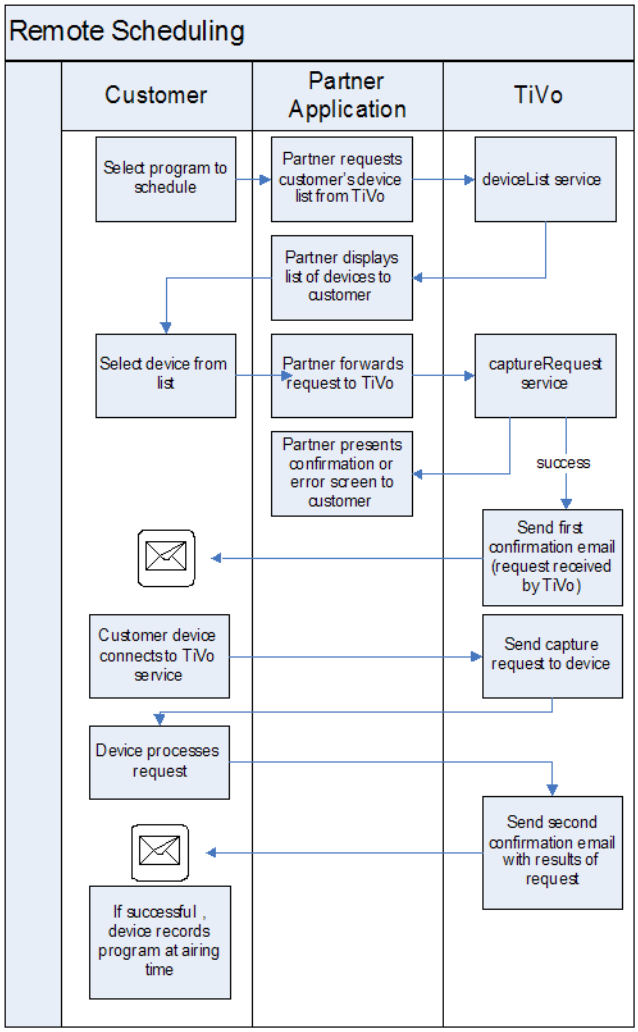
A TiVo DVR can receive scheduling requests when it connects to the TiVo service.

TiVo and its partners can deliver remote scheduling capabilities by using the `scheduleRecordingStore` operation. The response to this operation indicates whether the

request was received by the TiVo service, *not* whether the target device processed the request. Therefore, partners should encourage customers to request email confirmation from TiVo when their device has actually processed the request.

Recordings are created by the scheduler or the sync engine. For recordings that are recording standard broadcast television (type "stream"), the `startTime/endTime` fields should always be set: the scheduler sets the "requested" and "scheduled" fields when the recording is scheduled, and the device sets the "actual" fields based on the times the recording actually started and ended.

The following diagram summarizes the flow between a customer, partner, and TiVo during the remote scheduling process:




## 8.1.3.2 Input Parameters

The `scheduleRecordingStore` operation requires a set of `scheduleRecording` fields that describe the recording to schedule. For more information, see *scheduleRecording Field Group* on page 133.

### Sample XML Input

The following listings show sample XML input for a `scheduleRecordingStore` operation.

```
<scheduleRecordingStore>
  <channelNumber>1111</channelNumber>
  <email>asdf@example.com</email>
  <endTimePadding>120</endTimePadding>
  <keepTime>5</keepTime>
  <maxRecordings>4</maxRecordings>
  <partnerContentId>epgProvider:ct.EP0191774755</partnerContentId>
  <partnerCustomerId>test</partnerCustomerId>
  <recordingPriority>high</recordingPriority>
  <recordingType>explicit</recordingType>
  <startTimePadding>60</startTimePadding>
  <tivoSerialNumber>123456789098765</tivoSerialNumber>
</scheduleRecordingStore>
```

### Sample URL Request

The following URL invokes the `scheduleRecording` operation.

```
https://otwsqa.tivo.com/
scheduleRecordingStore?type=scheduleRecordingStore
&tivoSerialNumber=123456
&partnerCustomerId=test
&channelNumber=1111&email=asdf@yahoo.com
&endTimePadding=120
&keepTime=5&maxRecordings=4
&recordingType=explicit
&startTimePadding=60
&partnerContentId=epgprovider:ct.123456
&partnerStationId=epgprovider:st.1234
&recordingPriority=high
```

## 8.1.3.3 Output Element

The `scheduleRecording` operation returns a `scheduleRecordingResponse` element or an `error` element. Only if the operation triggers an error does the response consist of an `error` element. For more information about the `scheduleRecordingResponse` element, see *scheduleRecordingResponse* on page 135.

**Note:** Optional elements (such as `showStart`, `duration`, or `programTitle`) appear in the response only if they were specified in the request.

### Sample XML Response

The following listing shows a `scheduleRecordingResponse` for a successful `scheduleRecording` operation.

```
<scheduleRecordingResponse>
```

```
  <requestId>abcdef</requestId>
  <scheduleRecording>
    <channelNumber>1111</channelNumber>
    <partnerContentId>epgProvider:ct.EP0191774755
      </partnerContentId>
    <partnerStationId>epgprovider:st.1234</partnerStationId>
    <endTimePadding>120</endTimePadding>
    <keepTime>5</keepTime>
    <levelOfDetail>high</levelOfDetail>
    <maxRecordings>4</maxRecordings>
    <partnerCustomerId>test</partnerCustomerId>
    <recordingPriority>high</recordingPriority>
    <scheduleRecordingId>CaptureRequestId</scheduleRecordingId>
    <recordingType>explicit</recordingType>
    <startTimePadding>60</startTimePadding>
    <tivoSerialNumber>74683742372741243</tivoSerialNumber>
    <type>scheduleRecording</type>
  </scheduleRecording>
  <status>success</status>
</scheduleRecordingResponse>
```

# XML Objects

9

These XML objects may be used as parameters or returned by one or more web services.

## 9.1 activation Field Group

The `activation` Field Group wraps a `contract` element. That element specifies the services to activate on a device.

### Use

Input parameter, `serviceActivate`.

| Field Name | Description | Access |
|---|---|---|
| contract | For detailed information about the `contract` element, see *contract* on page 103 | Write |

## 9.2 address

The `address` element encapsulates fields that define a customer mailing address. The `address` element is always encapsulated in a `customer` element. For more information about the `customer` element, see *customer* on page 106.

### Use

Sub-element of the `customer` element input parameter for `customerInfoStore` and `serviceActivate`.

Returned as a sub-element of the `customer` element output in `customerInfoSearchResponse` and `deviceInfoSearchResponse`.

| Field Name | Description | Access |
|---|---|---|
| city | City name. Up to 40 characters.<br>Type: string | Read/Write |
| country | Country name. Up to 3 characters.<br>Type: string | Read/Write |
| postalCode | Postal or ZIP code. Up to 10 characters.<br>Type: string | Read/Write |

| Field Name | Description | Access |
|---|---|---|
| state | State code. Up to 3 characters.<br>Type: string | Read/Write |
| street1 | First address line. Up to 60 characters.<br>Type: string | Read/Write |
| street2 | Second address line. Up to 40 characters.<br>Type: string | Read/Write |

# 9.3 appError

Error information element. Returned by some operations as an output element if an error occurs during a service operation. The `appError` element that describes the error and the ID of the request that generated the error.

**Note:** Also see *error* on page 117.

For more information about output elements in general, including `error` elements, see *Receiving a Response* on page 26.

## Use

Output element for some unsuccessful operations.

| Field Name | Description | Access |
|---|---|---|
| code | Error code.<br>Type: string | Read |
| requestId | Identifies the request that generated the error.<br>Type: string | Read |
| text | Error message.<br>Type: string | Read |

# 9.4 availableCorrelatedCollectionForCollectionId

The `availableCorrelatedCollectionForCollectionID` element is a sub-element of the `content` element. It encapsulates fields that describe a collection that is related to a specific `contentSearch` operation. For more information about the content element, see *content* on page 99. For more information about `contentSearch`, see *contentSearch Operation* on page 80.

## Use

If the content returned by a `contentSearch` operation has related content, the `content` element wraps a single `availableCorrelatedCollectionForCollectionId` element for each piece of related content.

**Note:** To specify that a `contentSearch` should return `availableCorrelatedCollectionForCollectionID` information if any exists, include the optional `note` field in the `contentSearch` input parameter, and set its value to `availableCorrelatedCollectionForCollectionId`.

| Field Name | Description | Access |
|---|---|---|
| collectionId | The globally unique ID for a collection. Collection objects retrieved from the mind will always have a collection ID. Collections that are created by other parties and are included in content or recording objects may not have collection ids.<br>Type: id:collection. Example: tivo:cl.15566705 | Read |
| collectionType | One of: movie, playlist, series, song, special, webVideo, or other choices added in the future.<br>Type: string | Read |
| episodic | This flag defines whether or not a series is episodic. If it is a 1 or not set it is assumed to be an episodic series. If it is a 0 then it is a non-episodic series such as a movie or special.<br>Type: string | Read |
| objectIdAndType | Limits the search to an object with exactly the same value in this field.<br>Type: long. Example value: 344147194561071 | Read |
| title | The primary title by which this collection is known.<br>Type: string | Read |

# 9.5 callInfo

The `callInfo` element encapsulates device-specific call information. It is the main sub-element of the `callInfoSearchResponseList` element generated by the `callInfoSearch` operation.

A `callInfoSearch` operation returns configuration information captured from a specified device each time the device called the TiVo service during an optionally specified number of days. Each such call is encapsulated by a separate `callInfo` element.

For more information about `callInfoSearchResponseList`, see *callInfoSearchResponseList* on page 93.

## Use

Output element, a sub-element of the `callInfoSearchResposneList` element, `callInfoSearch` operation.

| Field Name | Description | Access |
|---|---|---|
| callInformation Field Group | *callInformation Field Group* on page 92 | Read |
| levelOfDetail | *levelOfDetail* on page 121 | Read |

# 9.6 callInformation Field Group

The `callInformation` field group details the main fields of the `callInfo` element. The `callInfo` element encapsulates device-specific call information in the `callInfoSearchResponseList` element generated by the `callInfoSearch` operation. For more information about `callInfo`, see *callInfo* on page 91.

| Field Name | Description | Access |
|---|---|---|
| ani | Automatic Number Identification. Applicable only for dial-up connections.<br>Type: string | Read |
| areaCodeSelected | Area code of the TiVo Service dial-in number selected during Guided Setup. Applicable only for dial-up connections.<br>Type: string | Read |
| callTime | Time of device call to the service.<br>Example: `11-11-2009 11:00:20`<br>Call time is in GMT and in a 24 hour format.<br>Type: string | Read |
| callType | Type of call. First setup call, daily call, software update call, or ping (test) call.<br>Examples: `First Setup`, `Second Setup`, `Daily`, `Daily and SW`, or `Ping`<br>Type: string | Read |
| configInfo | Configuration details.<br>See *configInfo* on page 94 | Read |
| dialInNumberSelected | TiVo Service dial-in number selected during Guided Setup. Applicable only for dial-up connections.<br>Type: string | Read |
| duration | Length of call in minutes. Applicable only for dial-up connections.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| zipcode | Zip or postal code of device location.<br>Type: string | Read |

# 9.7 callInfoSearchResponseList

The `callInfoSearchResponseList` element is returned by a successful `callInfoSearch` operation. It encapsulates one or more `callInfo` sub-elements.

## Use

Output element, `callInfoSearch`.

| Field Name | Description | Acess |
|---|---|---|
| callInfo | Call info: See *callInfo* on page 91 | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.8 channel

The `channel` element is a sub-element of the `recording` element, a sub-element of the `recordingList` or `recordingGroup` elements returned by a successful `recordingSearch` operation. The fields in `channel` provide information about the channel that was recorded. For more information about `recording`, see *recording* on page 125. For more information about `recordingList`, see *recordingList* on page 128. For information about recordingGroup, see *recordingGroup* on page 128.

## Use

Output element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| callSign | For recordings that were made via broadcast television, this is the call sign of the station that the recording was broadcast on when recorded. Example: KSBW.<br>Type: string. Maximum length: 32 characters. | Read |
| Number | The channel number, in major-minor format. Examples: 23, 30, 4-3.<br>Type: string. | Read |
| partnerstationId | A unique ID for the partner channel's station. Type: string. | Read |

| Field Name | Description | Access |
|---|---|---|
| sourceType | The signal source type. One of: cable, lineInput, satellite, terrestrial, unknown, or other choices added in the future.<br>Type: string. | Read |
| stationId | A unique ID for the channel's station (a TiVoId) Type: string. Example: tivo:st.454 | Read |
| type | Type: string. Example: channel | Read |

# 9.9 configInfo

The `configInfo` element encapsulates device-specific configuration information. It is the main sub-element in the `configInfoSearchResponse` element returned by a successful `configInfoSearch` operation. It is also a sub-element of the `device` and `callInfo` elements.

For more information about `configInfoSearch`, see *configInfoSearch Operation* on page 56.

## Use

Input parameter, a sub-element of the `device` element, `deviceInfoStore` operation.

Output element, `configInfoSearchResponse` element, `configInfoSearch` operation.

Output element, sub-element of the `device` element, `deviceInfoSearch` operation.

Output element, sub-element of the `callInfo` element, `callInfoSearch` operation.

| Field Name | Description | Access |
|---|---|---|
| configInfo Field Group | *configInfo Field Group* on page 94 | Read<br>(Write, deviceInfoStore only) |
| levelOfDetail | *levelOfDetail* on page 121 | Read |

# 9.10 configInfo Field Group

The `configInfo` field group details the main fields of the `configInfo` element. The `configInfo` element encapsulates device-specific configuration information in the `configInfoSearchResponse`, `callInfo`, and `device` elements.

The `configInfoSearchResponse` element is returned by a successful `configInfoSearch` operation. For more information about `configInfoSearchResponse`, see *configInfoSearchResponse* on page 98. For more information about `configInfoSearch`, see *configInfoSearch Operation* on page 56.

A `callInfo` element is returned by a successful `callInfoSearch` operation. For more information about `callInfo`, see *callInfo* on page 91. For more information about `callInfoSearch`, see *callInfoSearch Operation* on page 53.

The `device` element is used in several operations. For more information about `device`, see *device* on page 110.

**Note:** When the `configInfo` sub-element is provided as part of a parent `device` element in an input parameter (for example, as in the `deviceInfoStore` operation), some fields in `configInfo` are writeable as indicated in the following table.

| Field Name | Description | Access |
|---|---|---|
| activeUIMode | Active UI mode: `HDUI` (hi-def) or `SDUI` (standard def).<br>Type: string | Read |
| audioSource | Audio source on the device. Example: `Main`<br>Type: string | Read |
| broadband | True/False indicator that the DVR is connected to the TiVo service through broadband. One of: True, False.<br>Type: string | Read |
| cds | Content Delivery System. Enables content downloads. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| cdsEnabled | Content Delivery System. Enable/disable video downloads of 3rd party video content.device?<br>One of: `Yes`, `No`<br>Type: string | Read |
| controllerId | Indicates the value of the controller ID associated with the box. TiVo's auto-provisioning process uses controller ID to assign the correct headend service group.<br>**Note**: Available only with web services version 9. | Read (version 9) |
| dialup | Dialup<br>See *dialup* on page 115 | Read |
| drive | Drive<br>See *drive* on page 116 | Read |
| drive1SN | First internal drive serial number.<br>Type: string | Read |
| drive2SN | Second internal drive serial number, if applicable. This serial number is from the HD manufacturer.<br>Type: string | Read |
| dualTuner | Indicates the number of tuners in the device.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| enableTiVoSuggestions | Enable TiVo Suggestions flag (Yes/No). Indicates whether TiVo Suggestions is enabled. TiVo Suggestions are personalized shows selected to match your preferences.<br>Type: string | Read |
| esata | Enable support for an external eSATA storage device. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| executeScriptName | The name of a script to be run on the device for specific purposes, such as when a patch is needed. The name is known only when the device has the script.<br>Type: string | Read |
| forceBackhaul | Enable extra logging from the device to the TiVo Service for debugging purposes. This feature can be enabled manually for a limited time only. It cannot be enabled from a web service.<br>One of: true, false.<br>Type: string | Read |
| hardwareSerialNumber | Serial number from the manufacturer. Identifies partner-manufactured devices.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| headendName | Name associated with the headend.<br>**Note**: Available only with web services version 9. | Read (version 9) |
| hme | Home Media Engine. Enables access to broadband apps. One of: Y, N.<br>Type: string | Read<br>Write: deviceInfoStore only |
| hostId1 | ID of cable card slot 1.Type: string | Read |
| hostId2 | ID of cable card slot 2.Type: string | Read |
| httpAccess | Allows HTTP access to the device. One of: Y, N.<br>Type: string | Read<br>Write: deviceInfoStore only |
| inputSource | Input Source<br>See *inputSource* on page 120 | Read |
| irDBVersion | The latest version of the IR code database the device has, represented as a decimal value.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| lowerCard | Lower cable card ID<br>Type: string | Read |
| mrv | Enables the transfer of content from one DVR device to another DVR device. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| musicPhoto | Share music and photos from the PC and enables the Music & Photos menu item in the TiVo classic (SD) User Interface. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| network | Network<br>See *network* on page 122 | Read |
| pgdToDate | The latest date of PGD (program guide data) the device has.<br>Example: 12-02-2010<br>Type: string | Read |
| plantId | Indicates the value of the plant ID associated with the box. TiVo's auto-provisioning process uses plant ID to assign the correct headend service group.<br>**Note**: Available only with web services version 9. | Read (version 9) |
| recordQuality | Recording quality. Examples: `Low`, `Medium`, `High`, or `Best`<br>Type: string | Read |
| serviceState | Reported service state on the device. See *Service State* on page 30.<br>Type: string | Read |
| siteId | Partner Site ID for VOD (Video On Demand) location.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| swUpgradeAuthorized | Is the device authorized for software upgrade? One of: true, false.<br>Type: string | Read |
| swVersion | Device software version or build number.<br>Type: string | Read |
| swVersionGroup | Device software version release group. This will be the string to identify the version of TiVo software on the box.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| tco | TiVo Central Online. Enables recordings to be scheduled remotely. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| terminationCause | Who terminated the call. Example: `Recorder`<br>Type: string | Read |
| tfaInCC | TFA in CC:(800 authorization). Toll Free Authorization. Applicable only for dial-up connections.<br>Type: string | Read |
| timeZone | The time zone that the device is in. Example: `GMT-08:00`<br>Type: string | Read |
| tivoSerialNumber | 15-digit hexadecimal string. Unique for each device.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| ttcb | TTCB (TiVo To Come Back). Enables transfer of content from the TiVo Desktop PC application, or from the Internet via the TiVo Desktop, to a DVR device. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| ttg | TTG (TiVo To-Go). Enables transfer of content from a DVR device to a PC using the TiVo Desktop. One of: Y, N.<br>Type: string | Read<br>(Write: deviceInfoStore only) |
| tuningAdapterManufacturer | Tuning Adapter Manufacturer<br>Type: string | Read |
| upperCard | Upper cable card ID<br>Type: string | Read |

# 9.11 configInfoSearchResponse

The `configInfoSearchResponse` element is returned by a successful `configInfoSearch` operation. It encapsulates configuration information about a requested device, indicates the status of the operation, and provides the ID of the request.

## Use

Output element, `configInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| configInfo | Config info See *configInfo* on page 94 | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.12 content

The `content` element represents meta-data for one piece of content returned by a `contentSearch` operation. A successful `contentSearch` operation returns a `contentSearchResponse` element that contains a `contentList` element comprised of one or more `content` elements.

Each `content` element is associated with one primary collection. For TV shows, the primary collection is the series to which it belongs. A `content` element includes the fields from its primary collection.

For more information about `contentSearch`, see *contentSearch Operation* on page 80. For more information about `contentSearchResponse`, see *contentSearchResponse* on page 102. For more information about `contentList`, see *contentList* on page 101.

## Use

Output element, `contentSearch`.

| Field | Description | Access |
|---|---|---|
| autoOverriddenContentId | If this service has determined that this content overrides another, this field is used to hold the original contentId. This field is intended for use by U.S. broadband content providers who author their own content objects instead of using the ones provided by the U.S. EPG Provider (Tribune); other partners can ignore it. Also, if overriddenContentId is set, this field is ignored.<br>Type: string | Read |
| availableCorrelatedCollectionForCollectionId | If a content search returns information about related content, the response will contain one availableCorrelatedCollectionForCollectionId element for each related content item.<br>Type: availableCorrelatedCollectionForCollectionId | Read |
| collectionDescription | Description of the collection.<br>Type: string | Read |

| Field | Description | Access |
|---|---|---|
| collectionId | This is the TiVo-assigned unique identifier for this collection. Collection objects retrieved from the mind will always have a collection id. Collections that are created by other parties and are included in content or recording objects may not have collection ids.<br>Type: id:collection. Example: tivo:cl.15566705 | Read |
| collectionType | One of: movie, playlist, series, song, special, webVideo, or other choices added in the future.<br>Type: string | Read |
| contentId | This is the TiVo-assigned unique identifier for this content.<br>Type: id:content. Example: tivo:ct.33082087 | Read |
| contentType | If the value is not specified when the object is stored, the mind will set the value to the default value, and return the default value when the object is retrieved.<br>Default: video | Read |
| description | Description of the content.<br>Type: string | Read |
| episodeNum | The episode number of the content. This field has a maxOccurs=unbounded, because sometimes episodes are split up into multiple parts, each with their own episode numbers and then re-broadcast as a single episode. The rebroadcast will contain the episode numbers of all of the parts.<br>Type: int | Read |
| episodic | This flag defines whether or not a series is episodic. If it is a 1 or not set it is assumed to be an episodic series. If it is a 0 then it is a non-episodic series such as a movie or special.<br>Type: boolean | Read |
| imageUrl | Image URL associated with this collection.<br>Type: string | Read |
| isEpisode | This flag is set when this program object refers to a specific episode, a specific content stream. When set, every showing of this program has exactly the same content. Sometimes there are programs (like *All My Children*) where there is no information about each episode, and one content object is used for all episodes; such programs do not have this flag set. If this flag is set, TiVo will not record this program more than once, because recording the same content twice would just waste disk space.<br>Type: boolean | Read |

| Field | Description | Access |
|---|---|---|
| partnerContentId | The partner's own content Id.<br>Type: string | Read |
| partnerId | Id of the partner that created the object.<br>Type: string | Read |
| seasonNumber | For episodic TV Series, this field indicates the season of which this episode is a part.<br>Type: int | Read |
| subtitle | A secondary title by which this specific content (as distinguished from the collection) is known (e.g. episode title). May apply only to content with ambiguous primary titles.<br>Type: string | Read |
| title | The primary title by which this collection is known.<br>Type: string | Read |

# 9.13 contentList

A successful `contentSearch` operation returns a `contentSearchResponse` element that contains a `contentList` element comprised of one or more `content` elements, one for each piece of content found.

For more information about `contentSearch`, see *contentSearch Operation* on page 80. For more information about `contentSearchResponse`, see *contentSearchResponse* on page 102. For more information about `content`, see *content* on page 99.

## Use

Output element, `contenSearch`.

| Field Name | Description | Access |
|---|---|---|
| content | Type: content | Read |

# 9.14 contentSearch Field Group

The `contentSearch` field group describes the individual elements in XML input for a `contentSearch` operation. The `contentSearch` operation searches for and returns program details for a specific show or series, and optionally returns program details for similar shows. This operation is used in tivo.com for the "If you like this…" feature.

Use the fields in `contentSearch` to specify the device to search on, the specific content to search for, the type of content, and optionally, to set a flag to search for related content as well.

For more information about the contentSearch operation, see *contentSearch Operation* on page 80.

## Use

Input element, `contentSearch` operation.

| Field Name | Description | Access |
|---|---|---|
| collectionType | Static string.<br>Value= “series”<br>Type: string | Write |
| note | Static string<br>Value=”availableCorrelatedCollectionForCollectionId”<br>Type: string | Write |
| partnerContentId | Partner program ID of the program to record.<br>Format is "epgprovider:ct.xxxx".<br>Type: string | Write |
| tivoSerialNumber | 15-digit hexadecimal string. Unique for each device.<br>Type: string | Write |

# contentSearchResponse

The `contentSearchResponse` element encapsulates the successful results from a `contentSearch` operation. It returns the request ID, the status of the operation, and a `contentList` structure that contains one or more `content` elements describing individual pieces of content that match the search criteria.

If the search criteria included the optional `note` element, then `contentSearch` includes information about shows related to the specified search content, if any is found.

For more information about `contentSearch`, see *contentSearch Operation* on page 80. For more information about `contentList`, see *contentList* on page 101. For more information about `content`, see *content* on page 99.

## Use

Output element, `contentSearch`.

| Field Name | Description | Access |
|---|---|---|
| contentList | Type: contentList | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.15 contract

The `contract` element encapsulates information about a customer contract that identifies a device associated with the customer, the terms of service, customer-identifying details, billing data, and credit card and gift card information.

When `contract` is used as an input parameter its fields are writeable. When `contract` is returned as an operation response, fields are read-only.

## Use

Input parameter, `serviceActivate`

Input parameter, `serviceCancel`

Output element, `customerInfoSearchResponse`

Output element, `deviceInfoSearchResponse`

| Field Name | Description | Access |
|---|---|---|
| billToCustomer | Bill To Customer.<br>See *customer* on page 106 | Read |
| contract Field Group | *contract Field Group* on page 103 | Read/Write |
| creditCard | Credit card information.<br>See *creditCard* on page 105 | Read |
| customer | Ship To Customer.<br>See *customer* on page 106 | Read/Write |
| device | Device description.<br>See *device* on page 110 | Read/Write |
| giftCard | Reference to a gift card. Gift Card Pin number.<br>See *giftCard* on page 118 | Read |
| levelOfDetail | *levelOfDetail* on page 121 | Read/Write |
| service | Service description.<br>See *service* on page 135 | Read/Write |
| soldToCustomer | Sold To Customer.<br>See *customer* on page 106 | Read |

# 9.16 contract Field Group

The `contract` field group details the main fields specific to the `contract` element. The `contract` element encapsulates customer-specific contract information in the `configInfoSearchResponse`, `callInfo`, and `device` elements.

When `contract` is used as an input parameter its fields are writeable. When `contract` is returned as an operation response, fields are read-only.

| Field Name | Description | Access |
|---|---|---|
| dealerId | Dealer id.<br>Type: string | Read |
| dealerName | Dealer name.<br>Type: string | Read |
| id | Contract Id.<br>Type: string | Read |
| promoCode | Promotion code.<br>Type: string | Read |
| rewardSignup | Flag indicating the user signed up to rewards.<br>Type: boolean | Read |
| serviceState | Reported service state on the device. See *Service State* on page 30.<br>Type: string | Read |
| startDate | Date when the contract was created.<br>Type: date | Read |
| status | status of the contract.<br>Type: string | Read |
| warranty | Warranty SKU.<br>See *service* on page 135 | Read/Write |

# 9.17 contractList

A successful `customerInfoSearch` operation returns a `customerInfoSearchResponse` element that contains both a `contractList` element and a `customerContractList` element comprised of one or more `contract` elements, one for each contract found for a given customer.

For more information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36. For more information about `customerInfoSearchResponse`, see *customerContractList* on page 107. For more information about `contract`, see *contract* on page 103.

## Use

Output element, `customerInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| contract | Customer contracts. See *contract* on page 103 | Read |

# 9.18 creditCard

The `creditCard` element is a sub-element of the `contract` element. The `creditCard` element details the customer credit card information associated with a customer contract. For more information about `contract`, see *contract* on page 103.

## Use

Sub-element, `contract`.

| Field Name | Description | Access |
|---|---|---|
| creditCard Field Group | *creditCard Field Group* on page 105 | Read |
| levelOfDetail | *levelOfDetail* on page 121 | Read |

# 9.19 creditCard Field Group

The `creditCard` field group details the fields belonging to the `creditCard` element. These details fields define credit card information that is associated with a customer contract. For more information about `contract`, see *contract* on page 103.

| Field Name | Description | Access |
|---|---|---|
| address | Credit Card mailing address. See *address* on page 89 | Read |
| authCode | Credit Card authorization code.<br>Type: string | Read |
| authAmount | Credit Card authorization code. See *error* on page 117 | Read |
| creditCardType | Type of credit card.<br>Type: enum, one of:<br>• MASTERCARD<br>• MAST<br>• VISA<br>• AMEX<br>• DINERS_CLUB<br>• DISCOVER<br>• DISC<br>• ENROUTE<br>• JBC | Read |
| cvvCode | Credit Card cvv code.<br>Type: string | Read |
| expirationDate | Credit Card expiration date.<br>Type: string | Read |
| expirationMonth | Credit Card expiration Month.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| expirationYear | Credit Card expiration Year.<br>Type: string | Read |
| name | Name on the credit card.<br>Type: string | Read |
| number | Credit Card Number.<br>Type: string | Read |

# 9.20 customer

The `customer` element provides fields that detail customer information.New customer records are created and existing customer records modified using the `customerInfoStore` operation. Parts of the customer element are subsequently used in the `serviceActivate` operation as input, and  as output in response to a `customerInfoSearch` operation.

For detailed information about the fields in `customer`, see *customer Field Group* on page 106.

For more information about `customerInfoStore`, see *customerInfoStore Operation* on page 42. For more information about `serviceActivate`, see *serviceActivate Operation* on page 45. For more information about `customerInfoSearch` see *customerInfoSearch Operation*  on page 36.

## Use

Input parameter, `customerInfoStore`.

Input parameter, `serviceActivate`.

Output element, `customerInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| address | Address. See *address* on page 89 | Read/Write |
| customer Field Group | *customer Field Group* on page 106 | Read/Write |
| levelOfDetail | *levelOfDetail* on page 121 | Read/Write |

# 9.21 customer Field Group

The `customer` field group details the fields belonging to the `customer` element. These details fields define customer information that is associated with a customer contract. For more information abou `customer`, see *customer* on page 106.

| Field Name | Description | Access |
|---|---|---|
| customerContractList | Service Contract List. See *contractList* on page 104 | Read/Write |
| email | Email address. Up to 241 characters.<br>Type: string | Read/Write |

| Field Name | Description | Access |
|---|---|---|
| firstName | Up to 40 characters.<br>Type: string | Read/Write |
| lastName | Up to 40 characters.<br>Type: string | Read/Write |
| id | Customer ID.<br>Type: string | Read/Write |
| newPartnerCustomerId | Up to 20 characters. Sets a new Partner Customer ID for the customer. Should be unique for each customer.<br>Type: String | Read/Write |
| optStatus | Privacy status. See *optStatus* on page 124 | Read/Write |
| partnerCustomerId | Up to 20 characters. Identifies a customer within the TiVo system. Should be unique for each customer.<br>Type: string | Read/Write |
| partnerId | partner id.<br>Type: string | Read |
| phone | Up to 30 characters.<br>Type: string | Read/Write |

# 9.22 customerContractList

See *contractList* on page 104.

# 9.23 customerInfoSearchResponse

The `customerInfoSearchResponse` element encapsulates the information returned by a successful call to the `customerInfoSearch` operation, including a `customerList` sub-element that encapsulates one or more `customer` sub-elements, each of which provide information about a single customer that matches specified search criteria.

For more information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36. For information about `customerList`, see *customerList* on page 108. For more information about `customer`, see *customer* on page 106.

## Use

Output element, `customerInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| customerList | Customer list. See *customerList* on page 108 | Read |
| requestId | Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| status | Type: string | Read |

# 9.24 customerInfoStoreResponse

The `customerInfoStoreResponse` element encapsulates the information returned by a successful call to the `customerInfoStore` operation.

For more information about `customerInfoStore`, see *customerInfoStore Operation* on page 42.

## Use

Output element, `customerInfoStore`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.25 customerList

The `customerList` is returned on a successful `customerInforSearch` operation in a `customerInfoSearchResponse` element. The `customerList` element encapsulates one or more `customer` sub-elements, each of which provide information about a single customer that matches specified search criteria in a call to `customerInfoSearch`.

For more information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36. For more information about `customerInfoSearchResponse`, see *customerInfoSearchResponse* on page 107. For more information about `customer`, see *customer* on page 106.

## Use

Output element, `customerInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| customer | Customer list. See *customer* on page 106 | Read |

# 9.26 customerRequest Field Group

The `customerRequest` field group describes the identifying fields used in a `customerInfoSearch` operation to specify a customer for about which to retrieve account information. It includes the customer's device list, media access key, and privacy status.

A `customerRequest` field group must include at least one unique customer-identifying piece of information, either a unique partner or TiVo ID for the customer, or either a TiVo serial number or a hardware serial number identiying a customer device. The operation automatically filters result using the fields provided in the `customerRequest` field group. For example, if the request specifies `tivoCustomerId` and `tivoSerialNumber` (TSN), the return result only includes the customer and the contract associated with the given TSN.

The `customerRequest` field group can also include an optional `contractStatus` field indicating the type of customer records to return, either all active contracts for the customer, or all canceled customer contracts. If the optional `contractStatus` field is omitted, `customerInfoSearch` returns all contracts, active and canceled, associated with the specified customer.

For more information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36.

## Use

Input parameter, `customerInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| contractStatus | Status of the contract. Length: 1 character<br>Type: string | Write |
| hardwareSerialNumber | Serial number from the manufacturer. Identifies partner-manufactured devices.<br>Type: string | Write |
| partnerCustomerId | Up to 20 characters. Identifies a customer within the TiVo system. Should be unique for each customer.<br>Type: string | Write |
| tivoCustomerId | TiVo's unique identifier for the customer. Length: 10 characters.<br>Type: string | Write |
| tivoSerialNumber | 15-digit hexadecimal string. Unique for each device.<br>Type: string | Write |

# 9.27 customerStoreRequest Field Group

The `customerStoreRequest` field group encapsulates a `customer` element for the `customerInfoStore` operation. The `customerInfoStore` operation creates or modifies a customer record.

For more information about the `customer` element, see *customer* on page 106. For more information about `customerInfoStore`, see *customerInfoStore Operation* on page 42.

### Use

Input element, `customerInfoStore`.

| Field Name | Description | Access |
|---|---|---|
| customer | See *customer* on page 106 | Write |

## 9.28 date

TiVo internal data type, used by some operations to represent a date.

| Field Name | Description | Access |
|---|---|---|
| date | Specifies a date value.<br>Type: date | Read/Write |

## 9.29 dateTime

TiVo internal data type.=, used by some operations to represent a specific date and time.

| Field Name | Description | Access |
|---|---|---|
| dateTime | Specifies a date and time value.<br>Type: date/time | Read/Write |

## 9.30 device

A `device` element encapsulates information about a particular DVR. In actual use, such as when `device` is a sub-element of an operational response, only some fields may needed or returned.

For an overview of the `device` entity, see *About the Device Entity* on page 29

### Use

Input parameter, `deviceInfoStore`. For more information about `deviceInfoStore`, see *deviceInfoStore Operation* on page 63.

Input parameter, `serviceGroupFetch`. For more information about `serviceGroupFetch`, see *serviceGroupFetch Operation* on page 61.

Input parameter, output element, `serviceGroupStore`. For more information about `serviceGroupStore`, see *serviceGroupStore Operation* on page 69.

Input parameter, output element, `serviceGroupRemove`. For more information about `serviceGroupRemove`, see *serviceGroupRemove Operation* on page 67.

Output element, `deviceMessageSearchResponse`, `serviceCancelByCustomerIdResponse`. For more information about `serviceCancelByCustomerId`, see *serviceCancelByCustomerId Operation* on page 48.

Sub-element, `contract`. For more information about `contract`, see *contract* on page 103.

| Field Name | Description | Access |
|---|---|---|
| deviceFieldGroup | *deviceFieldGroup* on page 111 | Read/Write |
| levelOfDetail | *levelOfDetail* on page 121 | Read/Write |

# 9.31 deviceFieldGroup

The `device` field group details the fields belonging to the `device` element that contain device information associated with a particular DVR. For more information about `device`, see *device* on page 110.

| Field Name | Description | Access |
|---|---|---|
| callerID | Enables the Caller ID menu option to enable disable Caller ID.<br>Type: string | Read/Write |
| contract | Contract details. See *contract* on page 103 | Read/Write |
| configInfo | Configuration details. See *configInfo* on page 94 | Read/Write |
| deviceType | Device type. Example: TiVo Gemini DVR 80 hour.<br>Type: string | Read |
| encodingType | See *encodingType Enum* on page 117 | Read |
| firstCallDate | Date of first connection to the TiVo service.<br>Type: string | Read |
| hardwareSerialNumber | Serial number from the manufacturer. Identifies partner-manufactured devices.<br>Type: string | Read/Write |
| HDUI | Enables the HDUI menu option to enable/disable HD menus.<br>Type: string | Read/Write |
| hideAdult | Enables VOD Hide Adult menu option to enable/disable the Hide Adult feature.<br>Note: only applies to Arris VOD clients. | Read/Write |
| mak | Media Access Key<br>Type: string | Read |
| MRS | Enables streaming of content from one DVR device to another.<br>Type: string | Read/Write |

| Field Name | Description | Access |
|---|---|---|
| msoServiceID | msoServiceID specifies the MSO headend instance used to determine configuration parameters for SDV, VPN, and PPV services.<br>Type: string, maximum length: 125 alphanumeric and symbol characters | Read/Write |
| name | Name of the device. Up to 16 characters.<br>Type: string | Read/Write |
| overrideAutoDiscovery | Allows manual provisioning of a box although auto discovery is supported. Reserved for future use. | Read/Write |
| partnerId | Unique identifier for the partner.<br>Type: string | Read |
| ppv | Enables Impulse Pay Per View. | Read/Write |
| purchasePin | Enables VOD Purchase Pin | Read/Write |
| sdv | Enables switched digital video. | Read/Write |
| siteId | Site ID specifies the MSO VOD site instance provisioned to the device.<br>Type: string | Read/Write |
| subPartnerId | Unique identifier for the sub partner. This is applicable if a partner is going to sell the box to a sub-partner.<br>Type: string | Read |
| tier | Used to determine the enabled features for this TCD.<br>Type: string | Read/Write |
| tivoSerialNumber | 15-digit hexadecimal number. Unique for each device.<br>Type: string | Read/Write |
| tuneToApp | Enables direct tuning to the VOD catalog (channel 1 only). | Read/Write |
| vod | Reserved for future use. | Read/Write |
| vpn | Enables VPN, which other MSO services may rely on. Partners should check with their MSO service deployment team (PPV, SDV, etc.) to determine if this setting is required and which services depend on it. | Read/Write |

# 9.32 deviceInfo

The `deviceInfo` element is returned as a sub-element of the `deviceInfoSearchResponse` element. It encapsulates information about the device, the contract associated with the device, and configuration information about the device.

For more information about `deviceInfoSearchResponse`, see *deviceInfoSearchResponse* on page 113.

## Use

Output element, `deviceInfoSearchResponse`.

| Field Name | Description | Access |
|---|---|---|
| configInfo | *configInfo* on page 94 | Read |
| contract | *contract* on page 103 | Read |
| device Field Group | *deviceFieldGroup* on page 111 | Read |
| levelOfDetail | *levelOfDetail* on page 121 | Read |

# 9.33 deviceInfoSearchResponse

The `deviceInfoSearchResponse` element is returned by a successful `deviceInfoSearch` operation. It encapsulates configuration information about a requested device, indicates the status of the operation, and provides the ID of the request. With version 9 web services, it also returns manufacturing information.

For more information about `deviceInfoSearch`, see *deviceInfoSearch Operation* on page 58.

## Use

Output element, `deviceInfoSearch`.

| Field Name | Description | Access |
|---|---|---|
| deviceInfo | Device info. See *device* on page 110 | Read |
| manufacturingInfo | Manufacturing information. See *manufacturingInfo* on page 122. | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.34 deviceInfoStoreResponse

The `deviceInfoStoreResponse` element is returned by a successful `deviceInfoStore` operation. It indicates the status of the operation, and provides the ID of the request.

For more information about `deviceInfoStore`, see *deviceInfoStore Operation* on page 63.

## Use

Output element, `deviceInfoStore`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| status | Type: string | Read |

# 9.35 deviceMessageSendResponse

The `deviceMessageSendResponse` element is returned inside a `deviceMessageSendResponseList` element by a successful `deviceMessageSend` operation. The `deviceMessageSendResponse` element encapsulates configuration information about a single device to which an MBM or PTCM is sent, and  indicates the status of the operation.

For more information about `deviceMessageSend`, see *deviceMessageSend Operation* on page 77. For more information about `deviceMessageSendResponseList`, see *deviceMessageSendResponseList* on page 114.

## Use

Output element, `deviceMessageSend`.

| Field Name | Description | Access |
|---|---|---|
| device | See *device* on page 110 | Read |
| status | Type: string | Read |

# 9.36 deviceMessageSendResponseList

The `deviceMessageSendResponseList` element is returned by a successful or partially successful `deviceMessageSend` operation. It encapsulates one or more `deviceMessageSendResponse` elements, each of which provides information about one device to which a message is sent, and the status of the operation on that device. The `deviceMessageSendResponseList` element also provides the ID of the send message request.

For more information about `deviceMessageSend`, see *deviceMessageSend Operation* on page 77. For more information about `deviceMessageSendResposne`, see *deviceMessageSendResponse* on page 114.

## Use

Output element, `deviceMessageSend`.

| Field Name | Description | Access |
|---|---|---|
| deviceMessageSendResponse | See *deviceMessageSendResponse* on page 114 | Read |
| requestId | Type: string | Read |

# 9.37 dialCodeUpdateResponse

The `dialCodeUpdateResponse` element is returned by a successful `dialCodeUpdate` operation. It indicates the status of the operation, and provides the ID of the request.

## Use

Output element. `dialCodeUpdate`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.38 dialup

The `dialup` element is a sub-element of the `configInfo` element. It provides information about the dialup connection to a device, including the call-waiting prefix, dial prefix, dial-tone check, phone availability, and pulse dial information. For more information about these `dialup` fields, see *dialupInfo Field Group* on page 115.

For more information about `configInfo`, see *configInfo* on page 94.

## Use

Sub-element, `configInfo`.

| Field Name | Description | Access |
|---|---|---|
| dialupInfo Field Group | *dialupInfo Field Group* on page 115 | Read |
| levelOfDetail | *levelOfDetail* on page 121 | Read |

# 9.39 dialupInfo Field Group

The `dialupInfo` field group details the fields belonging to the `dialup` element. These details fields describe call-waiting prefix, dial prefix, dial-tone check, phone availability, and pulse dial information for a device. For more information about `dialup`, see *dialup* on page 115.

| Field Name | Description | Access |
|---|---|---|
| callWaitingPrefix | Call Waiting Prefix.<br>Type: string | Read |
| dialPrefix | Dial Prefix.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| dialToneCheck | Dial Tone Check.<br>Type: string | Read |
| phoneAvailabilityDetection | Phone Availability Detection.<br>Type: string | Read |
| pulseDial | Tone/Pulse Dial.<br>Type: string | Read |

# 9.40 drive

The `drive` element is a sub-element of the `configInfo` element. It provides information about the hard drive associated with a device, including the drive model, serial number, firmware, size, connection, and whether the drive is supported or not.

For more information about `configInfo`, see *configInfo* on page 94.

## Use

Sub-element, `configInfo`.

| Field Name | Description | Access |
|---|---|---|
| driveModel | Model of external drive.<br>Type: string | Read |
| driveSize | Size of external drive.<br>Type: string | Read |
| firmware | Firmware of external drive.<br>Type: string | Read |
| married | External drive indicator. Values:<br>• 1 - Device is connected to external HD.<br>• 0 - Device is not connected to external HD.<br>Type: string | Read |
| serialNumber | Serial Number of external drive.<br>Type: string | Read |
| supported | Support indicator. Values:<br>• YES - Supported. The device does not support connecting to an external HD.<br>• NO - Not supported.<br>Type: string | Read |

# 9.41 encodingType Enum

A field in the `device` field group that specifies the encoding employed by a device. One of:

- mpeg2ProgramStream
- mpeg2TransportStream
- vc1ApL3
- avcL41MP4

For more information about the `device` field group, see *deviceFieldGroup* on page 111.

# 9.42 endTimePadding

A field in the `scheduleRecording` field group that specifies the number of seconds after the end time of the showing to continue recording. It is used to compensate for stations that do not keep exact time.

The `scheduleRecording` field group is used in `scheduleRecordingStore`, and it is returned in `scheduleRecordingResponse`. For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85. For more information about `scheduleRecordingResponse`, see *scheduleRecordingResponse* on page 135.

Allowed values: 0, 60, 120, 180, 240, 300, 600, 900, 1800, 3600, 5400, 10800.

**Note:** If `endTimePadding` is omitted from a call to `scheduleRecordingStore`, the service automatically sets the value of `endTimePadding` to 0 meaning no padding occurs.

# 9.43 error

Error information element. Returned by most operations as an output element if an error occurs during a service operation. The `error` element describes the error and the ID of the request that generated the error.

For more information about output elements, including `error` elements, see *Receiving a Response* on page 26.

**Note:** Also see *appError* on page 90.

## Use

Output element, service operation errors.

| Field Name | Description | Access |
|---|---|---|
| code | Error code.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| requestId | Identifies the request that generated the error.<br>Type: string | Read |
| text | Error message.<br>Type: string | Read |

# 9.44 giftCard

A sub-element of the `contract` element, `giftCard` encapsulates information about a gift card associated with a particular customer contract. For a specific list of the fields provided for recording and reviewing gift card information, see *giftCard Field Group* on page 118.

## Use

Sub-element, `contract`.

| Field Name | Description | Access |
|---|---|---|
| giftCard Field Group | *giftCard Field Group* on page 118 | Read |
| levelOfDetail | *levelOfDetail* on page 121 | Read |

# 9.45 giftCard Field Group

The `giftCard` field group details the fields belonging to the `giftCard` element. For more information about `giftCard`, see *giftCard* on page 118.

| Field Name | Description | Access |
|---|---|---|
| amount | See *error* on page 117 | Read |
| dateTime | Type: string | Read |
| denomination | Type: string | Read |
| giftCardId | Redemption Code/pin number.<br>Type: string | Read |
| lineDesc | Type: string | Read |
| merchant | Type: string | Read |
| providerId | Type: string | Read |
| redemtionDateTime | Type: string | Read |
| response | See *Allowed values: 0, 60, 120, 180, 240, 300, 600.* on page 146 | Read |
| serialNumber | Serial Number.<br>Type: string | Read |
| sku | Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| status | Type: string | Read |
| store | Type: string | Read |
| storeAddress | See *address* on page 89 | Read |
| upc | Type: string | Read |
| vendor | Type: string | Read |

# 9.46 group

The `group` element defines a custom service group for use with the `serviceGroupFetch`, `serviceGroupRemove`, and `serviceGroupStore` operations. A custom group is defined by a group prefix (for example, DG for a distribution group), and a unique group name.

For more information about `serviceGroupFetch`, see *serviceGroupFetch Operation* on page 61. For more information about `serviceGroupRemove`, see *serviceGroupRemove Operation* on page 67. For more information about `serviceGroupStore`, see *serviceGroupStore Operation* on page 69.

## Use

Input parameter, `serviceGroupFetch`.

Input parameter, `serviceGroupRemove`.

Input parameter, `serviceGroupStore`.

Output element, `serviceGroupFetch`.

| Field Name | Description | Access |
|---|---|---|
| groupName | Maximum length: 32 characters.<br>Type: string | Read/Write |
| groupPrefix | One of: DG, AP, SF, DC, MC, SG | Read/Write |

# 9.47 id:collection

TiVo data type, used to define the `collectionID` field in the `content`, `recording`, and `availableCorrelatedCollectionForCollectionId` elements.

Format:: "tivo:cl.xxxx".

For more information about `content`, see *content* on page 99. For more information about `recording`, see *recording* on page 125. For more information about

availableCorrelatedCollectionForCollectionId, see *availableCorrelatedCollectionForCollectionId* on page 90.

| Field Name | Description | Access |
|---|---|---|
| id:collection | Type: string | Read/Write |

# id:content

TiVo data type, used to define the contentID field in the content and recording elements. It provides an identifier for the content of the program to record.

Format:: "tivo:ct.xxxx".

For more information about content, see *content* on page 99. For more information about recording, see *recording* on page 125.

| Field Name | Description | Access |
|---|---|---|
| id:content | Type: string | Read/Write |

# 9.48 inputSource

The inputSource element identifies the source from which a device receives programming, and is part of the data returned in the configInfoSearchResponseList element by a successful configInfoSearch operation or in the callInfoSearchResponse element by a successful callInfoSearch operation.

For a description of the fields in inputSource, see *inputSource Field Group* on page 121.

For more information about configInfoSearch, see *configInfoSearch Operation* on page 56. For more information about callInfoSearch, see *callInfoSearch Operation* on page 53. For more information about configInfoSearchResponse, see *configInfoSearchResponse* on page 98. For more information about callInfoSearchResponseList, see *callInfoSearchResponseList* on page 93.

## Use

Output element, configInfoSearchResponse element, configInfoSearch operation.

Output element, callInfoSearchResponse element, callInfoSearch operation.

| Field Name | Description | Access |
|---|---|---|
| inputSourceFieldGroup | *inputSource Field Group* on page 121 | Read |

# 9.49 inputSource Field Group

The `inputSource` field group details the fields belonging to the `inputSource` element, including the headend, lineup, and set top brand, if any. For more information, see *inputSource* on page 120.

| Field Name | Description | Access |
|---|---|---|
| headend | ID of the headend where incoming programming is received.<br>Type: string | Read |
| lineup | Name of the channel lineup group.<br>Type: string | Read |
| setTopBrand | Set Top Box Brand.<br>If a set top brand is not specified, the value in this field is set to “Empty”.<br>Type: string | Read |

# 9.50 levelOfDetail

The `levelOfDetail` field appears in results from many web service operations to indicate how much detail is reported. The `leveOfDetail` is always one of `high`, `medium`, or `low`.

## Use

Output element field.

| Field Name | Description | Access |
|---|---|---|
| levelOfDetail | One of: high, low, medium.<br>Type: string | Read |

# 9.51 maxRecordings

The `maxRecordings` field is part of the `scheduleRecording` field group. When used as an input field, it specifies the maximum number of episodes to schedule for recording. As an output field it indicates the number of episodes that are scheduled for recording.

The default value for `maxRecording` is 5. If `maxRecordings` is set to 0, all episodes are scheduled for recording. Other permissible values are 1, 2,3, 4, or 10.

## Use

Input field, `scheduleRecording`.

Output field, `scheduleRecordingResponse`.

# 9.52 manufacturingInfo

The `manufacturingInfo` element is returned as a sub-element of the `deviceInfoSearchResponse` element. It encapsulates information about the manufacturing of the device.

For more information about `deviceInfoSearchResponse`, see *deviceInfoSearchResponse* on page 113.

## Use

Output element, `deviceInfoSearchResponse`.

| Field Name | Description | Access |
|---|---|---|
| ccDate | Date code of the last update | Read |
| ccId | ID number of the cable card | Read |
| ccManufacturer | Name of the manufacturer of the cable card (CSCO or MOTO) | Read |
| ccSerialNumber | Serial number of cable card paired at time of manufacturing | Read |
| ccShipTo | Zip code of the site to which the unit will be shipped | Read |
| ccUnitAddress | Unit address of the cable card | Read |
| dataId | Scraped data ID string; only for MOTO cards | Read |
| hostId | Scraped host ID string | Read |
| levelOfDetail | Level of detail displayed | Read |
| macAddress | Scraped CableCard MAC string; optional for MOTO cards | Read |
| test5Date | Date the TCD completed testing at manufacturer | Read |
| tivoSerialNumber | TSN associated with the CableCard | Read |
| Type | | Read |

# 9.53 network

The `network` element is a sub-element of the `configInfoSearch` and `callInfoSearch` operations. On successful calls to these operations network is returned as a sub-element of `configInfoSearchResponse` and `callInfoSearchResponseList`, respectively.

The `network` element describes a device's network attributes.

For detailed information about the fields in the `network` element, see *networkInformation Field Group* on page 123.

For more information about `configInfoSearch`, see *configInfoSearch Operation* on page 56. For more information about `configInfoSearchResponse`, see *configInfoSearchResponse* on page 98.

For more information about `callInfoSearch`, see *callInfoSearch Operation* on page 53. For more information about `callInfoSearchResponseList`, see *callInfoSearchResponseList* on page 93.

## Use

Output element, `configInfoSearchResponse`.

Output element, `callInfoSearchResponseList`.

| Field Name | Description | Access |
|---|---|---|
| levelOfDetail | *levelOfDetail* on page 121 | Read |
| networkInformationFieldGroup | *networkInformation Field Group* on page 123 | Read |

# 9.54 networkInformation Field Group

The `networkInformation` field group details the fields belonging to the `network` element, including physical connection type, dynamic IP address, network type, network adapter, wireless signal strength, and the MAC address of the network adapter.

For more information about `network`, see *network* on page 122.

| Field Name | Description | Access |
|---|---|---|
| dynamicIPAddress | Is the device using a dynamic IP address? One of: `Yes`, `No`, or `No Info`.<br>Type: string | Read |
| macAddress | Network adaptor MAC address.<br>Type: string | Read |
| networkAdaptor | Network adaptor name.<br>Type: string | Read |
| peerToPeerNetwork | Indicator if the DVR is connected through a peer-to-peer network or an adhoc network. One of `Yes`, `No`, or `No Info`.<br>Type: string | Read |
| physicalConnectionType | Physical connection type. Example: `Wired`, `Wireless`, or `No Info`.<br>`Wired` connects using physical CAT wire. `Wireless` connects using wireless adaptor.<br>Type: string | Read |

| Field Name | Description | Access |
|---|---|---|
| wirelessSignalStrength | Indicator of the signal strength of the wireless network to which the device is connected.<br>Type: string | Read |

# 9.55 optStatus

The `optStatus` element describes a customer's chosen level of privacy. It is a sub-element of the `customer` element. A customer's `optStatus` can be set by using the `customerInfoStore` operation to create or modify a customer record. A customer's `optStatus` is reported by successful `customerInfoSearch` and `deviceInfoSearch` operations in the `customerInfoSearchResponse` and `deviceInfoSearchResponse` elements, respectively.

**Note:** To view the TiVo Privacy Policy on the web, visit: http://www.tivo.com/abouttivo/policies/tivoprivacypolicy.html.

For more information about the `customer` element, see *customer* on page 106.

For more information about `customerInfoStore`, see *customerInfoStore Operation* on page 42. For more information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36. For more information about `deviceInfoSearch`, see *deviceInfoSearch Operation* on page 58.

## Use

Input sub-element of `customer` element, `customerInfoStore`.

Output sub-element of `customer` element, `customerInfoSearch`.

Outout sub-element of `customer` element, `deviceInfoSearch`.

| Level | Description | Access |
|---|---|---|
| optIn | Allow collection of personally identifiable viewing data for Audience Research and Measurement | Read/Write |
| optNeutral | Allow collection of anonymous personal viewing data for Audience Research and Measurement | Read/Write |
| optOut | Prevent collection of all viewing data, personal and anonymous | Read/Write |

**Note:** When `optStatus` is set to `optOut`, the following features that depend on collecting personal data for operation are unavailable: Online services, tivo.com: View My Shows List, View To Do List, Season Pass Manager, Remote Scheduling, TiVo Suggestions, HDUI features such as Search and browse, broadband applications and streaming features (such as Amazon, YouTube, Netflix), and Discovery Bar recommendations based on thumbs ratings.

# 9.56 recording

The `recording` element is a sub-element of the `recordingList` and `recordingGroup` elements that are returned by a successful `recordingSearch` operation.

A `recording` element encapsulates information about a discrete piece of recorded content. When a `recordingSearchResponse` contains multiple entries in a `recordingList` or `recordingGroup`, each content entry is detailed in a separate `recording` sub-element.

For more information about `recordingSearch`, see *recordingSearch Operation* on page 83.

For more information about `recordingSearchResponse`, see *recordingSearchResponse* on page 131. For more information about `recordingList`, see *recordingList* on page 128. For more information about `recordingGroup`, see *recordingGroup* on page 128.

## Use

Output element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| channel | The channel from which the recording was captured.<br>Type: channel | Read |
| collectionId | This is the TiVo-assigned unique identifier for this collection. Type: id:collection. Example: tivo:cl.15566705 | Read |
| contentId | This is the TiVo-assigned unique identifier for this content. Type: id:contentId. Example: tivo:ct.33082087 | Read |
| contentType | What format the content is in.<br>If the value is not specified when the object is stored, the mind will set the value to the default value, and return the default value when the object is retrieved.<br>Type: string. Default: video | Read |
| deletionPolicy | If this is set to `whenSpaceNeeded`, the recording can be deleted when space is needed.<br>If this is set to `neverDelete`, the body should attempt to not delete this recording. If this is not set, the default value of whenSpaceNeeded should be used.<br>In the case it's set to `specificDate`, the date the user chose will be specified in the desiredDeletion field.<br>This recording may be required to be deleted prior to the date specified here for DRM requirements. This will be specified in the `drm` attribute.<br>Type: string | Read |
| duration | Recording duration in seconds.<br>Type: int | Read |

| Field Name | Description | Access |
|---|---|---|
| episodeNum | The episode number of the recording.<br>Type: int | Read |
| episodic | This flag defines whether or not a series is episodic. If it is a 1 or not set it is assumed to be an episodic series. If it is a 0 then it is a nonepisodic series such as a movie or special.<br>Type: boolean | Read |
| expectedDeletion | When is this content expected to be deleted. This may happen for many reasons. If there is no expected expiration time, this element will not be included.<br>Type: dateTime | Read |
| hdtv | This flag indicates whether this particular recording is high definition, which we define as vertical video resolution greater than 576.<br>Type: boolean | Read |
| isEpisode | This flag is set when this program object refers to a specific "episode", a specific content stream. When set, every showing of this program has exactly the same content. Sometimes there are programs (like All My Children) where there is no info about each episode, and one content object is used for all episodes; such programs do not have this flag set.<br>If this flag is set, TiVo will not record this program more than once, because recording the same content twice would just waste disk space.<br>Type: boolean | Read |
| mimeType | What format the content is in.<br>Type: string. Default: video/mpg2 | Read |
| partnerCollectionId | Partner series ID of the program. Format is "epgprovider:cl.xxxx".<br>Type: string | Read |
| partnerContentId | Partner program ID of the program.<br>Format is "epgprovider:ct.xxxx".<br>Type: string | Read |
| recordingId | Identifies the recording.<br>Type: string. Example: tivo:rc.3329 | Read |
| requestedEndPadding | The number of seconds of end padding that the user requested when setting up the recording. May be different that the scheduled padding because of trade-offs made by the scheduler.<br>Type: int | Read |

| Field Name | Description | Access |
|---|---|---|
| requestedStartPadding | The number of seconds of start padding that the user requested when setting up the recording. May be different that the scheduled padding because of trade-offs made by the scheduler.<br>Type: int | Read |
| seasonNumber | For episodic TV Series, this field indicates the season of which this episode is a part.<br>Type: int. Minimum value: 0 | Read |
| startTime | For offers that are broadcast television (type "stream"), this field should always be set to the startTime of the offer: the scheduler sets it when the recording is scheduled, and the body updates it based on the actual time. For CDS downloads, it is never set: the scheduler doesn't say when it should start, and the body doesn't set it based on actual times. For manual recordings, startTime is the time the recording starts (NOT the startTime of the offer)<br>Type: dateTime (format is YYYY-MM-DD HH:MM:SS in GMT, ex: 2009-02-04 20:00:00) | Read |
| state | The state of this recording, according to the body. One of: cancelled, complete, contentDeleted, deleted, inProgress, pending, reset, scheduled, or other choices added in the future.<br>Type: string | Read |
| subtitle | A secondary title by which this specific content (as distinguished from the collection) is known (e.g. episode title). May apply only to content with ambiguous primary titles.<br>If a value that exceeds the maximum length is supplied, it will be truncated without returning an error.<br>Type: string. Maximum length: 255 characters | Read |
| title | The primary title by which this collection is known. If a value that exceeds the maximum length is supplied, it will be truncated without returning an error.<br>Type: string. Maximum length: 255 characters | Read |
| tivoSerialNumber | 15-digit hexadecimal string. Unique for each device. | Read |
| type | Content type.<br>Type: string. Example: recording | Read |

# 9.57 recordingGroup

The `recordingGroup` element is a sub-element of the `recordingGroupList` element returned in a `recordingSearchResponse` by a successful `recordingSearch` operation when the `recordingSearch` request includes the optional `groupBy` parameter.

A `recordingGroup` element contains a count of the number of recordings in the group, and one recording element for each content recording.

For more information about `recordingSearch`, see *recordingSearch Operation* on page 83. For more information about `recordingGroupList`, see *recordingGroupList* on page 128.

## Use

Output element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| count | The number of recordings in the group.<br>Type: int | Read |
| recording | A recording object for each recording returned by the search. | Read |

# 9.58 recordingGroupList

The `recordingGroupList` element is a sub-element of `recordingSearchResponse`, returned by a successful `recordingSearch` operation when the `recordingSearch` request includes the optional `groupBy` parameter.

A `recordingGroupList` contains one or more `recordingGroup` sub-elements.

For more information about `recordingSearch`, see *recordingSearch Operation* on page 83. For more information about `recordingGroup`, see *recordingGroup* on page 128.

## Use

Output element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| recordingGroup | Recording group | Read |

# 9.59 recordingList

The `recordingList` element is a sub-element of `recordingSearchResponse`, returned by a successful `recordingSearch` operation.

A `recordingList` contains one or more `recording` sub-elements, each of which encapsulates information about a discrete piece of recorded content.

For more information about `recordingSearch`, see *recordingSearch Operation* on page 83. For more information about `recording`, see *recording* on page 125.

### Use

Output element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| recording | A recording object for each recording returned by a recordingSearch operation. | Read |

## 9.60 recordingPriority

`recordingPriority` is both a field in the `scheduleRecording` element of a `scheduleRecordingStore` operation and a field in the `scheduleRecordingResponse` element returned by a successful `scheduleRecordingStore` operation. It indicates requested or assigned priority level for the recording.

For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85.

### Use

Input field, `scheduleRecordingStore`.

Output element, `scheduleRecordingStore`.

| Field Name | Description | Access |
|---|---|---|
| recordingPriority | Indicates the recording priority for a scheduled recording.<br>One of:<br>• low (default)<br>• high<br>Type: string | Read/Write |

## 9.61 recordingQualityLevel

`recordingQualityLevel` is a field in the `scheduleRecording` element that indicates the desired level of recording quality in a `scheduleRecordingStore` operation. For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85.

## Use

Input field, `scheduleRecordingStore`.

| Field Name | Description | Access |
|---|---|---|
| recordingQualityLevel | One of:<br>• best<br>• high<br>• medium (default)<br>• good<br>Type: string | Write |

# 9.62 recordingSearch

The `recordingSearch` element encapsulates a request for a `recordingSearch` operation. For more information about `recordingSearch`, see *recordingSearch Operation* on page 83.

## Use

Input element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| count | The maximum number of recordings to return. If the value is not specified when the object is stored, the mind will set the value to the default value, and return the default value when the object is retrieved.<br>Type: int, default: 1, minimum value: 1, maximum value: 50 | Write |
| groupBy | Optional. When set, the results of the query will be group objects, where all objects with the same value in the grouped field are together.<br>Type: string. If provided, the value must be “collectionId”. (A collectionId value is a series identifier.) | Write |
| hardwareSerialNumber | Serial number from the manufacturer. Identifies partner-manufactured devices.<br>Type: string | Write |
| minStartTime | Returns offers whose startTime is greater than or equal to minStartTime<br>Type: dateTime (format is YYYY-MM-DD HH:MM:SS in GMT, for example, 2009-02-04 20:00:00) | Write |

| Field Name | Description | Access |
|---|---|---|
| offset | The offset from the anchor. If not specified, default is 0.<br>If the value is not specified when the object is stored, the mind will set the value to the default value, and return the default value when the object is retrieved.<br>Type: int, default: 0 | Write |
| orderBy | Specifies the sort order for the returned objects.<br>Valid values:<br>deletionTime, isInProgress, isSuggestion, startTime, subtitle, title<br>If more than one value is specified, the first defines the primary sort order, the second specifies the order for objects that match in the first field, and so on.<br>Type: string. Maximum length: 32 characters. | Write |
| state | One of: cancelled, complete, contentDeleted, deleted, inProgress, pending, reset, scheduled, or other choices added in the future.<br>Specify complete for recordings in the Now Playing List, specify scheduled for recordings in the To Do List.<br>Type: string | Write |
| tivoSerialNumber | 15-digit hexadecimal string. Unique for each device.<br>Type: string | Write |

# 9.63 recordingSearchResponse

The `recordingSearchResponse` element is returned by a successful `recordingSearch` operation. It indicates the ID of the initial `recordingSearch` request, the status of the operation, and it provides either a `recordingList` or `recordingGroupList` that contains information about one or more recordings that matched the search criteria.

By default a `recordingSearchResponse` contains a `recordingList` element unless the initial `recordingSearch` request includes the optional `groupBy` parameter. If the `groupBy` parameter is provided, `recordingSearchResponse` contains a `recordingGroupList` instead of a `recordingList`.

For more information about `recordingSearch` and `groupBy` see *recordingSearch Operation* on page 83.

For more information about `recordingList` see *recordingList* on page 128. For more information about `recordingGroupList`, see *recordingGroupList* on page 128.

### Use

Output element, `recordingSearch`.

| Field Name | Description | Access |
|---|---|---|
| recordingList or recordingGroupList | Type: `recordingList` or `recordingGroupList` | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

## 9.64 recordingType

`recordingType` is both a field in the `scheduleRecording` element of a `scheduleRecordingStore` operation and a field in the `scheduleRecordingResponse` element returned by a successful `scheduleRecordingStore` operation. It indicates the kind of recording requested.

For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85.

### Use

Input field, `scheduleRecordingStore`.

Output element, `scheduleRecordingStore`.

| Field Name | Description | Access |
|---|---|---|
| recordingType | One of:<br>• explicit<br>• seasonPass (default)<br>Type: string | Read/Write |

## 9.65 scheduleRecording

The `scheduleRecording` element both describes a recording request for the `scheduleRecordingStore` operation and is returned in a `scheduleRecordingResponse` element by a successful `scheduleRecordingStore` operation.

For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85. For a detailed description of the fields belonging to the `scheduleRecording` element, see *scheduleRecording Field Group* on page 133. For more information about `scheduleRecordingResponse`, see *scheduleRecordingResponse* on page 135.

## Use

Input parameter, `scheduleRecordingStore`.

| Field Name | Description | Access |
|---|---|---|
| levelOfDetail | *levelOfDetail* on page 121 | Write |
| scheduleRecording Field Group | *scheduleRecording Field Group* on page 133 | Write |

# 9.66 scheduleRecording Field Group

The `scheduleRecording` field group details the fields belonging to the `scheduleRecording` element. For more information about the `scheduleRecording` element, see *scheduleRecording* on page 132.

| Field Name | Description | Access |
|---|---|---|
| aclFlag | Internal field. Determines whether partner can access the service or not.<br>Type: string | Read |
| anonId | Internal field. Unique ID for customer.<br>Type: string | Read |
| channelNumber | Same channel number as in the line-up.<br>Type: string | Write |
| contentId | This is the TiVo-assigned unique identifier for this content. Format is "tivo:ct.xxxx".<br>See *id:content* on page 120 | Write |
| duration | Duration of the program in seconds. Optional.<br>Type: string | Write |
| email | Email address to send recording notification.<br>Type: string | Write |
| endTimePadding | Number of seconds after the end time of the showing to continue recording. This allows the viewer to compensate when stations don't keep exact time. If negative, it makes the recording end early. If missing, acts as though it were 0.<br>See *endTimePadding* on page 117 | Write |
| firstEmailFlag | One of: yes, no. A value of "no" means "Don't send the email". The default value is "no".<br>Type: string | Write |
| hardwareSerialNumber | Serial number from the manufacturer. Identifies partner-manufactured devices.<br>Type: string | Write |

| Field Name | Description | Access |
|---|---|---|
| keepTime | The number of days to keep the recording on the device. -1 means until I delete, 0 means keep until space needed. Values from 1-7 mean the number of days. Default is 0.<br>Type: int | Write |
| maxRecordings | The maximum number of recordings to keep at any one time. Should be greater than or equal to 1. If missing, means unlimited.<br>See *maxRecordings* on page 121 | Write |
| namespace | Identifier (partner name) used to define the required fields for the partner.<br>Type: string | Write |
| partnerContentId | The partner's own content ID.<br>Type: string | Write |
| partnerCustomerId | Up to 20 characters. Identifies a customer within the TiVo system. Should be unique for each customer.<br>Type: string | Write |
| partnerEnv | Identifier used for connecting to staging or production Engineering mind specific to the partner.<br>Type: string | Write |
| partnerId | Internal field. Unique Id for the partner.<br>Type: string | Read |
| partnerStationId | The partner's own station ID.<br>Type: string | Write |
| programTitle | The primary title by which the corresponding program is known. Optional.<br>Type: string | Write |
| recordingId | Recording ID generated for this scheduling. It will be sent in the response.<br>Type: string | Read |
| recordingPriority | Sets the recording priority. One of: low, high. Default: low<br>See *recordingPriority* on page 129 | Write |
| recordingQualityLevel | The quality setting for recordings from this scheduleRecording. One of: best, high, medium, good. Default: medium.<br>See *recordingQualityLevel* on page 129 | Write |
| recordingType | See *recordingType* on page 132 | Write |
| showStart | The air date time of the program in GMT. Optional.<br>Type: string | Write |

| Field Name | Description | Access |
|---|---|---|
| showStatus | See showType.<br>See *showType* on page 145 | Write |
| startTimePadding | Number of seconds before the actual start of the showing to begin recording. This allows the viewer to compensate when stations don't keep exact time. If negative, it makes the recording start late. If missing, acts as though it were 0.<br>See *startTimePadding* on page 146 | Write |
| stationId | Identifies the station on which the program is being aired. The format is "tivo:st.xxxx" where xxxx represents the partner-specific station ID.<br>See *inputSource* on page 120 | Write |
| tivoSerialNumber | 15-digit hexadecimal string. Unique for each device.<br>Type: string | Write |

# 9.67 scheduleRecordingResponse

The `scheduleRecordingResponse` element is returned by a successful `scheduleRecordingStore` operation. It indicates the ID of the initial scheduleRecordingStore request, the status of the operation, and it provides a `scheduleRecording` sub-element detailing the successfully scheduled recording.

For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85. For more information about `scheduleRecording`, see *scheduleRecording* on page 132.

## Use

Output element, `scheduleRecordingStore`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| scheduleRecording | See *scheduleRecording* on page 132 | Read |
| status | Type: string | Read |

# 9.68 service

The `service` element encapsulates the services offered in a `contract` element used in the `serviceActivate`, `serviceCancel`, `customerInfoSearch`, and `deviceInfoSearch` operations.

For detailed information about the fields in the `service` element, see *service Field Group* on page 139. For more information about the `contract` element, see *contract* on page 103.

For information about `serviceActivate`, see *serviceActivate Operation* on page 45. For information about `serviceCancel`, see *serviceCancel Operation* on page 47. For information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36. For information about `deviceInfoSearch`, see *deviceInfoSearch Operation* on page 58.

### Use

Input parameter, `serviceActivate`.

Input parameter, `serviceCancel`.

Output element, `customerInfoSearchResponse`.

Output element, `deviceInfoSearchResponse`.

| Field Name | Description | Access |
|---|---|---|
| levelOfDetail | *levelOfDetail* on page 121 | Read/Write |
| service Field Group | *service Field Group* on page 139 | Read/Write |

## 9.69 serviceActivateResponse

The `serviceActivateResponse` element is returned by a successful `serviceActivate` operation. It indicates the status of the operation, the ID of the request, and provides details about the contract associated with the service activation.

For more information about `serviceActivate`, see *serviceActivate Operation* on page 45.

### Use

Output element, `serviceActivate`.

| Field Name | Description | Access |
|---|---|---|
| contract | Contract details. See *contract* on page 103 | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

## 9.70 serviceCancel Field Group

The `serviceCancel` field group details the fields belonging to the `serviceCancel` element for the `serviceCancel` operation. These fields include a contract sub-element that details the service to cancel on a device, and a cancellation code.

For more information about `serviceCancel`, see *serviceCancel Operation* on page 47.

### Use

Input element, `serviceCancel`.

| Field Name | Description | Access |
|---|---|---|
| cancellationCode | A fixed code that is provided to the partner by TiVo.<br>Type: string | Write |
| contract | Contract details. See *contract* on page 103 | Write |

## 9.71 serviceCancelByCustomerId Field Group

The `serviceCancelByCustomerId` field group details the fields belonging to the `serviceCancelByCustomerId` element for the `serviceCancelByCustomerId` operation. These fields include a cancellation code provided by TiVo, and either a partner ID for the customer whose service to cancel, or a TiVo customer ID. The supplied ID is used to determine the device for which to cancel service.

For more information about `serviceCancelByCustomerId`, see *serviceCancelByCustomerId Operation* on page 48.

### Use

Input element, `serviceCancelByCustomerId`.

| Field Name | Description | Access |
|---|---|---|
| cancellationCode | A fixed code that is provided to the partner by TiVo.<br>Type: string | Write |
| partnerCustomerId | Type: string | Write |
| tivoCustomerId | Type: string | Write |

## 9.72 serviceCancelByCustomerIdResponse

The `serviceCancelByCustomerIdResponse` element is a sub-element of the `serviceCancelByCustomerIdResponseList` returned by a successful or partially successful `serviceCancelByCustomerId` operation. Each device matching the service cancellation criteria is returned in its own `serviceCancelByCustomerIdResponse` sub-element within the `serviceCancelByCustomerIdResponseList`.

Each `serviceCancelByCustomerIdResponse` element indicates the device for which services were canceled, the status of the operation, and an error indication, if any.

For more information about `serviceCancelByCustomerId`, see *serviceCancelByCustomerId Operation* on page 48. For more information about `serviceCancelByCustomerIdResponseList`, see *serviceCancelByCustomerIdResponseList* on page 138.

## Use

Output element, `serviceCancelByCustomerId`

| Field Name | Description | Access |
|---|---|---|
| device | See *device* on page 110 | Read |
| error | See *appError* on page 90 | Read |
| status | Type: string | Read |

# 9.73 serviceCancelByCustomerIdResponseList

The `serviceCancelByCustomerIdResponseList` is returned by a successful or partially successful `serviceCancelByCustomerId` operation. It reports the ID of the processed request, and provides one or more `serviceCancelByCustomerIdResponse` sub-elements for each device on which service is canceled.

For more information about `serviceCancelByCustomerId`, see *serviceCancelByCustomerId Operation* on page 48. For more information about `serviceCancelByCustomerIdResponse`, see *serviceCancelByCustomerIdResponse* on page 137.

## Use

Output element, `serviceCancelByCustomerId.`

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| serviceCancelByCustomerIdResponse | See *serviceCancelByCustomerIdResponse* on page 137 | Read |

# 9.74 serviceCancelResponse

The `serviceCancelResponse` element is returned by a successful `serviceCancel` operation. It reports the ID of the processed request, and indicates the status of the operation.

## Use

Output element, `serviceCancel.`

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.75 service Field Group

The `service` field group details the fields in the `service` element, a sub-element of the `contract` element used in the `serviceActivate`, `serviceCancel`, `customerInfoSearch`, and `deviceInfoSearch` operations.

For more information about the `contract` element, see *contract* on page 103.

For information about `serviceActivate`, see *serviceActivate Operation* on page 45. For information about `serviceCancel`, see .*serviceCancel Operation* on page 47. For information about `customerInfoSearch`, see *customerInfoSearch Operation* on page 36. For information about `deviceInfoSearch`, see *deviceInfoSearch Operation* on page 58.

## Use

Input parameter, `serviceActivate`.

Input parameter, `serviceCancel`.

Output element, `customerInfoSearchResponse`.

Output element, `deviceInfoSearchResponse`.

| Field Name | Description | Access |
|---|---|---|
| chargePrice | Actual charge price of the service. See *error* on page 117 | Read |
| description | Name of the Service SKU. For example, “MSO Monthly Service”.<br>Type: string | Read |
| discountPrice | discount on the price of the service. See *error* on page 117 | Read |
| name | name of the service e.g Product Life Time or one year monthly.<br>Type: string | Read |
| price | Price of the service. See *error* on page 117 | Read |
| serviceType | Service Type.<br>Type: string | Read |
| sku | Service Id Generally service SKU in SAP.<br>Type: string | Read/Write |
| tax | Tax on the price of the service. See*error* on page 117 | Read |

# 9.76 serviceGroupFetchResponse

The `serviceGroupFetchResponse` element is returned by a successful `serviceGroupFetch` operation. It lists the requested group, the status of the operation, and the ID of the request.

For more information about `serviceGroupFetch`, see *serviceGroupFetch Operation* on page 61.

## Use

Output element, `serviceGroupFetch`.

| Field Name | Description | Access |
|---|---|---|
| group | *serviceGroupFetchResponse* on page 139 | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.77 serviceGroupRemoveResponse

The `serviceGroupRemoveResponse` element is a sub-element of a `serviceGroupRemoveResponseList` returned by a successful or partially successful `serviceGroupRemove` operation. Each `serviceGroupRemoveResponse` element in a `serviceGroupRemoveResponseList` provides data about the removal of a group on a single device. If removal fails on a device, the corresponding `serviceGroupRemoveResponse` element contains an `error` sub-element. Otherwise the `error` element is not present.

For more information about `serviceGroupRemove`, see *serviceGroupRemove Operation* on page 67. For more information about `serviceGroupRemoveResponseList`, see *serviceGroupRemoveResponseList* on page 140.

## Use

Output element, `serviceGroupRemove`.

| Field Name | Description | Access |
|---|---|---|
| device | See *device* on page 110. | Read |
| error | See *appError* on page 90 | Read |
| status | Type: string | Read |

# 9.78 serviceGroupRemoveResponseList

The `serviceGroupRemoveResponseList` element is returned by a successful or partially successful `serviceGroupRemove` operation. It provides the ID of the request, and one or more `serviceGroupRemoveResponse` sub-elements, one for each device for which an attempt was made to remove a service group. Each `serviceGroupRemoveResponse` describes the success or failure of the removal operation on a single device.

For more information about `serviceGroupRemove`, see *serviceGroupRemove Operation* on page 67. For more information about `serviceGroupRemoveResponse`, see *serviceGroupRemoveResponse* on page 140.

## Use

Output element, `serviceGroupRemove`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| serviceGroupRemoveResponse | See *serviceGroupRemoveResponse* on page 140 | Read |

# 9.79 serviceGroupStoreResponse

The `serviceGroupStoreResponse` element is a sub-element of a `serviceGroupStoreResponseList` returned by a successful or partially successful `serviceGroupStore` operation. Each `serviceGroupStoreResponse` element in a `serviceGroupStoreResponseList` provides data about the creation of a group on a single device. If creation fails on a device, the corresponding `serviceGroupStoreResponse` element contains an `error` sub-element. Otherwise the `error` element is not present.

For more information about `serviceGroupStore`, see *serviceGroupStore Operation* on page 69. For more information about `serviceGroupStoreResponseList`, see *serviceGroupStoreResponseList* on page 141.

## Use

Output element, `serviceGroupStore`.

| Field Name | Description | Access |
|---|---|---|
| device | See *device* on page 110. | Read |
| error | See *appError* on page 90 | Read |
| status | Type: string | Read |

# 9.80 serviceGroupStoreResponseList

The `serviceGroupStoreResponseList` element is returned by a successful or partially successful `serviceGroupStore` operation. It provides the ID of the request, and one or more `serviceGroupStoreResponse` sub-elements, one for each device for which an attempt was made to create a service group. Each `serviceGroupStoreResponse` describes the success or failure of the operation on a single device.

For more information about `serviceGroupStore`, see *serviceGroupStore Operation* on page 69. For more information about `serviceGroupStoreResponse`, see *serviceGroupStoreResponse* on page 141.

### Use

Output element, `serviceGroupStore`

| Field Name | Description | Access |
|---|---|---|
| requestID | Type: string | Read |
| serviceGroupStoreResponse | See *serviceGroupStoreResponse* on page 141 | Read |

## 9.81 serviceMessage Field Group

The `serviceMessage` field group details the information required to create or modify a message using the `serviceMessageStore` operation.

For more information about `serviceMessageStore`, see *serviceMessageStore Operation* on page 72.

### Use

Input parameter, `serviceMessageStore`.

| Field Name | Description | Access |
|---|---|---|
| expiryDate | The expiration date of the message. | Write |
| from | The sender of the message.<br>Type: string | Write |
| messageId | The ID of the message.<br>Type: string | Write |
| messageText | The text of the message.<br>Type: string | Write |
| messageType | Allowed values: ptcm or mbm | Write |
| subject | The subject of the message.<br>Type: string | Write |

## 9.82 serviceMessageList

The `serviceMessageList` element is part of a `serviceMessageSearchResposne` element that is returned by a successful `serviceMessageSearch` operation. A `serviceMessageList` consists of one or more `serviceMessage` sub-elements, each of which encapsulates the fields pertaining to a single message that matched the specified search criteria.

For more information about `serviceMessageSearch`, see *serviceMessageSearch Operation* on page 75. For more information about `serviceMessageSearchResponse`, see *serviceMessageSearchResponse* on page 143.

### Use

Output element, `serviceMessageSearch`.

| Field Name | Description | Access |
|---|---|---|
| serviceMessage | See *serviceMessage Field Group* on page 142. | Read |

## 9.83 serviceMessageSearch Field Group

The `serviceMessageSearch` field group encapsulates two input elements for a `serviceMessageSearch` operation: a message ID that identifies the specific message for which to search, and the status of messages to search for.

For mroe information about `serviceMessageSearch`, see *serviceMessageSearch Operation* on page 75.

### Use

Input parameter, `serviceMessageSearch`.

| Field Name | Description | Access |
|---|---|---|
| messageId | Identifies the message to search for. | Write |
| messageStatus | Search for messages that have the specified status.<br>Allowed values: Active, Expired | Write |

## 9.84 serviceMessageSearchResponse

The `serviceMessageSearchResponse` element is returned by a successful `serviceMessageSearch` operation. It provides the ID of the request, the status of the operation, and a `serviceMessageList` element containing one or more messages that matched the search criteria.

For more information about `serviceMessageSearch`, see *serviceMessageSearch Operation* on page 75. For more information about `serviceMessageList`, see *serviceMessageList* on page 142.

### Use

Output element, `serviceMessageSearch`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| serviceMessageList | One or more serviceMessages.<br>See *serviceMessage Field Group* on page 142. | Read |

| Field Name | Description | Access |
|---|---|---|
| status | Type: string | Read |

# 9.85 serviceMessageStoreResponse

The `serviceMessageStoreResponse` element is returned by a successful `serviceMessageStore` operation. It indicates the ID of the request, the status of the operation, and the ID of the newly created or modified message.

For more information about `serviceMessageStore`, see *serviceMessageStore Operation* on page 72.

## Use

Output element, `serviceMessageStore`.

| Field Name | Description | Access |
|---|---|---|
| messageId | The ID of the PTCM (Pre-Tivo Central Message) or MBM (Message Board Message) | Read |
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.86 serviceMessageRemoveResponse

The `serviceMessageRemoveResponse` element is returned by a successful `serviceMessageRemove` operation. It indicates the ID of the request and the status of the operation.

For more information about `serviceMessageRemove`, see *serviceMessageRemove Operation* on page 74.

## Use

Output element, `serviceMessageRemove`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| status | Type: string | Read |

# 9.87 serviceReset Field Group

The `serviceReset` field group encapsulates a `device` element sent as input to a `serviceReset` operation. It indicates the device to reset.

For more information about `serviceReset`, see *serviceReset Operation* on page 50. For more information about the `device` element, see *device* on page 110..

### Use

Input parameter, `serviceReset`.

| Field Name | Description | Access |
|---|---|---|
| device | Device. See *device* on page 110 | Write |

## 9.88 serviceResetResponse

The serviceResetResponse element is returned by a successful `serviceReset` operation. It indicates the ID of the request and the status of the operation.

For more information about `serviceReset`, see *serviceReset Operation* on page 50.

### Use

Output element, `serviceReset`.

| Field Name | Description | Access |
|---|---|---|
| requestId | Type: string | Read |
| status | Type: string | Read |

## 9.89 showType

The `showType` field is an element in the `scheduleRecording` field group. It specifies what versions of content to record.

**Note:** If the `recordingType` is also explicitly declared in `scheduleRecording`, then the value in this `showType` is ignored.

For more information about the scheduleRecording operation, see *scheduleRecording* on page 132. For more information about the scheduleRecording field group, see *scheduleRecording Field Group* on page 133.

## Use

Input element, `scheduleRecording`.

| Field Name | Description | Access |
|---|---|---|
| showType | One of the following allowed values:<br>• EVERY (record every episode, the default value for this element.<br>• FiRSTRUN (first run only)<br>• RERUNS (recording reruns allowed)<br>Type: string | Write |

# 9.90 startTimePadding

A field in the `scheduleRecording` field group that specifies the number of seconds before the start of a showing to begin recording. It is used to compensate for stations that do not keep exact time.

The `scheduleRecording` field group is used in `scheduleRecordingStore`, and it is returned in `scheduleRecordingResponse`. For more information about `scheduleRecordingStore`, see *scheduleRecordingStore Operation* on page 85. For more information about `scheduleRecordingResponse`, see *scheduleRecordingResponse* on page 135.

Allowed values: 0, 60, 120, 180, 240, 300, 600.

# Error Codes

10

This section lists error codes returned by TiVo IT Web Services.

The following table identifies error code ranges and the general meaning of error codes within those ranges.

| Code Ranges | Descriptions |
|---|---|
| 1000-19999 | Shared codes |
| 20000-29999 | Account service codes |
| 30000-39999 | Commerce service codes |
| 4000-4999 | Common service codes |
| 50000-59999 | Device service codes |
| 60000-69999 | Schedule service codes |

The following table identifies individual error codes and what they mean.

| Code | Description |
|---|---|
| 1000 | Invalid TiVo Serial Number. TiVo Serial Number should be a 15 digit alpha numeric. |
| 1001 | Internal exception. |
| 1002 | Internal exception. |
| 1003 | Connection to external system failed. |
| 1004 | Certificate Information not present in the request or an invalid certificate presented. |
| 1005 | Partner not authorized to do this operation. |
| 1006 | Partner not found in the system. |
| 1007 | Invalid email format. |
| 1008 | Invalid Hardware Serial Number. |
| 1009 | Partner Customer Id invalid. |
| 1010 | TiVo Serial Number not found. |
| 1011 | Hardware Serial Number not found. |
| 1012 | This TiVo Serial Number is not valid for the partner. |
| 1013 | badArgument: `<description>`. |
| 1014 | internalError: `<descripton>`. |
| 21001 | Customer Group is missing. |
| 21002 | Customer Group is not maintained in the SAP Master table. |
| 21003 | Input at least one reference field for getting the customer details. |
| 21004 | Customer not found for the given Partner Customer ID. |
| 21005 | Invalid Customer found for the Partner Customer ID. |

| Code | Description |
|---|---|
| 21006 | The TiVo Customer ID does not match the Partner Customer ID. |
| 21008 | No Valid Contracts. Please check the TiVo Serial Number. |
| 21009 | TiVo Serial Number is not related to an MSO Contract. |
| 21010 | Invalid Purchase Order No. |
| 21011 | Invalid PO number for the TiVo Serial Number. |
| 21012 | Invalid TiVo Customer ID. |
| 21013 | This TiVo Serial Number does not pertain to an MSO contract. |
| 21014 | This PO does not pertain to an MSO contract. |
| 21015 | No contract available for the TiVo Customer ID. |
| 21100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 22001 | Customer Group is missing. |
| 22002 | Customer Group is not maintained in the SAP Master table. |
| 22003 | The Partner Customer ID is missing. |
| 22004 | Invalid TiVo Customer ID. |
| 22005 | Multiple customers exist for the same email address. |
| 22007 | Location error in address. |
| 22010 | Customer not found for the given Partner Customer ID. |
| 22036 | The Partner Customer ID already exists. |
| 22100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 22600 | Exception: *Error message.* |
| 23001 | Error while sending the email. |
| 23002 | Customer not found. |
| 23100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 30004 | Customer not found for the given Partner Customer ID. |
| 30009 | Error while fetching partner name. |
| 30010 | PartnerId missing. |
| 31001 | Partner Customer ID is missing. |
| 31002 | Last Name is missing. |
| 31003 | Street1 is missing. |
| 31004 | Postal Code is missing. |
| 31005 | City is missing. |
| 31006 | State is missing. |
| 31007 | Location Error in address. |
| 31008 | Phone is missing. |
| 31009 | Customer Group is missing. |
| 31010 | SKU is missing. |
| 31011 | TiVo Serial Number is missing. |

| Code | Description |
|---|---|
| 31012 | Order Reason is missing. |
| 31013 | Invalid Partner ID or Customer Group or Order Reason. |
| 31014 | TiVo Serial Number not found. |
| 31016 | Input Email and Partner Customer ID do not match. |
| 31017 | This TiVo Serial Number has active contract. |
| 31018 | The TiVo Serial Number does not pertain to the MSO partner. |
| 31019 | No contract found for MSO Reactivation. |
| 31020 | Contract is created today. The TSN cannot be reactivated on the same day. |
| 31022 | The Partner Customer ID and the Customer ID do not match. |
| 31024 | Invalid Partner Customer ID for the Email Address submitted. |
| 31042 | This TiVo Serial Number does not pertain to the MSO partner. |
| 31043 | Email ID already assigned to a different MSO partner. |
| 31100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 31200 | Exception: *Error message.* |
| 31300 | Invalid SiteId. |
| 32001 | The Customer Group is missing. |
| 32002 | The Reason for Rejection code is missing. |
| 32003 | The TiVo Serial Number is missing. |
| 32004 | The Customer Group is not maintained in the SAP Master table. |
| 32005 | The Partner Customer ID is missing. |
| 32006 | The PO Number or Contract ID is missing. |
| 32007 | Invalid Customer Group. |
| 32008 | Invalid Rejection Code. |
| 32009 | Invalid Partner Customer ID. |
| 32010 | Invalid Contract ID. |
| 32011 | Invalid Purchase Order No. |
| 32012 | TiVo Serial Number not found. |
| 32015 | No Contracts found for the given TiVo Serial Number. |
| 32016 | The Contract is already cancelled. |
| 32017 | No Contracts found for the given TiVo Serial Number. |
| 32018 | The customer is not associated with this Partner Customer ID. |
| 32019 | Contract does not match the TiVo Serial Number contract. |
| 32020 | The input PO number does not match the input TiVo Serial Number contract. |
| 32022 | The input TiVo Serial Number does not pertain to the MSO Contract. |
| 32100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 32200 | Exception: *Error message.* |
| 33017 | This TSN has active contract: *contract number.* |

| Code | Description |
|---|---|
| 33101 | The TiVo Serial Number is not found. |
| 33111 | The TiVo Serial Number does not pertain to the partner. |
| 33100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 50000 | No Message ID is specified. |
| 50001 | No Message Name is specified. |
| 50002 | Send Device Command failed. Failure reason: <failure reason>. |
| 50003 | Unable to retrieve device configuration information. |
| 50004 | No contracts found. |
| 50005 | No device found. |
| 50006 | Device name cannot exceed 16 characters. |
| 50007 | Customer not found. |
| 50008 | Device Name or Flags must be provided as parameters. |
| 50009 | Error while fetching partner name. |
| 50011 | No call info found. |
| 50012 | Cannot send PTCMs or MBMs to this DVR unit from customer groups. |
| 50013 | PartnerId missing. |
| 50014 | Cannot find partner info from database using certificate information. |
| 50015 | `messageId` or `messageText` must be provided as parameters. |
| 50016 | tivoSerialNumbers, hardwareSerialNumbers, or both must be provided as parameters. |
| 50017 | Message not found. |
| 50018 | Device does not belong to a partner. |
| 50027 | Update could not be applied because of device tier. |
| 50028 | Can not update tier for this partner. |
| 50100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 50300 | Invalid SiteId. |
| 50301 | Receiver name not unique for customer. |
| 60000 | Schedule Recording failed. Failure reason: <failure reason>. |
| 60001 | Invalid email format. |
| 60009 | Error while fetching partner name. |
| 60010 | PartnerId missing. |
| 60100 | There are missing parameters or bad values in the request. Missing or invalid parameters. |
| 60101 | Invalid hardwareSerialNumber. |
| 60200 | Operation is not valid because of device tier. |

# Glossary

11

## Digital Video Recorder (DVR)

Digital video recorder. Industry term for a consumer electronics device that stores digital video. Basic features are the ability to pause and rewind Live TV, and automatically record programs.

## Media Access Key (MAK)

A Media Access Key (MAK) enables access TiVo recordings from a home network. Not to be confused with MAC (Media Access Control) an address that identifies a computer or other hardware device on a local area network.

## Service State

Each TiVo device is assigned a service state that changes during the device life cycle. For example, when a device is manufactured, it is initialized to a service state of 1 (New). When the TiVo service is activated, the service state is set to 3 (Good).

## TiVo Service Number (TSN)

TiVo Service Number. This is the 15-digit hexadecimal string which uniquely identifies a TiVo DVR.

## TiVo to Come Back (TTCB)

A feature of the TiVo Desktop software that transfers video files from a PC or other device to a TiVo DVR.

## TiVoToGo™ (TTG)

A feature of the TiVo Desktop software that transfers video files from a TiVo DVR to a PC or other device.

# Index

12

## A



## C



## D

## E



## F



## G



## I



## L



## M



## N



## O

## P



## R



## S

## T



## U



## V