

SMS: Recommendations

Ver.	Date	Reason for revision
00	2018	Initial version
01	09-10-2018	Update section 1 to add good reasons to follow recommendations

Table of content

1	Aim/objective	1
2	Scope	1
3	Connectivity	1
3.1	Permanent connection	1
3.2	Status polling	2
3.3	Timeout	2
4	Optimized workflow	2
4.1	Minimize number of calls	2
4.2	Limit refresh	2
5	Backward compatibility	2
5.1	New fields	2
5.2	Error codes	3

1 Aim/objective

This document provides recommended practices when integrating the Barco SMS API in a TMS or any other client application.

Following the recommendations below will:

- Avoid SMS log flooding issue;
- Improve system (TMS-SMS) performances;
- Avoid backward compatibility issues

2 Scope

This document is intended for software engineers.

3 Connectivity

3.1 Permanent connection

Barco strongly recommends API clients to maintain a single HTTPS connection open when monitoring the SMS status. It is also recommended to perform the Login command once and keep the session open.

The same remark applies to connections to other modules of the projector.

Opening and closing frequently the connection is not efficient and can flood the system log. This can impede a proper analysis of log.

SMS: Recommendations

3.2 Status polling

Barco recommends API clients to regularly call the GetMainStatus command to maintain the user session open and to monitor the status. It should not be called at more than 1 Hz.

A default 5 minutes timeout is set on user session but TCP connections may last less than that depending on the keep-alive settings. It is recommend calling GetMainStatus once every minute to keep the connection and login active.

The response of GetMainStatus provides a summary of the SMS status and last change times that indicates when some data have changed. This can help to refresh lists of content. However a TMS that modifies the content list (i.e. adding or deleting content and keys) should preferably refresh the content lists once all changes are done.

3.3 Timeout

Barco recommends API clients to set a response timeout to a minimum of 10 seconds. An API command may be delayed by other command calls or updates occurring at the same time on the SMS.

4 Optimized workflow

4.1 Minimize number of calls

Depending on its workflow, the client application should try to minimize the number of API calls by grouping them.

Some examples:

- it is more optimized to delete multiple CPLs in one call than deleting them one by one
- it is better to update the schedule for a range of one or several day(s) than to add show one by one

4.2 Limit refresh

If a client application is performing several consecutive actions on content or keys, it should avoid refreshing content lists after each change. It will be more efficient to refresh the changed data once all changes are applied.

5 Backward compatibility

The Barco API is backward compatible to some extents but require some flexibility on client side.

5.1 New fields

API clients should support parsing SOAP responses that would contain fields that were not present in the wsdl.

5.2 Error codes

The Barco API returns a set of error codes that may keep growing with new software releases even if the API version is not changed. Existing codes won't be changed but new ones can be added.

As functionalities are evolving, an existing command might need to return new error messages.

The `GetErrorMessageMap` command returns a list of codes with severity and a short text.