SLACK CLIENT FOR CRESTRON

Revision: 2.04
Date: 03 December 2021

SUMMARY

Ultamation's Slack Client module connects a Slack Bot account with a Crestron system. The module allows a Slack channel to be displayed and send new message on a Crestron device. Multiple Slack modules can be used at the same time to integrate more than one channel.

This Datasheet provides the essential information for integration between Slack and a Crestron program, and information on programming the module, with a host Crestron program.

This module is compatible with 3 and 4-Series Crestron systems ONLY.

We recommend testing the module prior to purchase by downloading the module and using the 1 hour trail period.

INSTALLATION NOTES

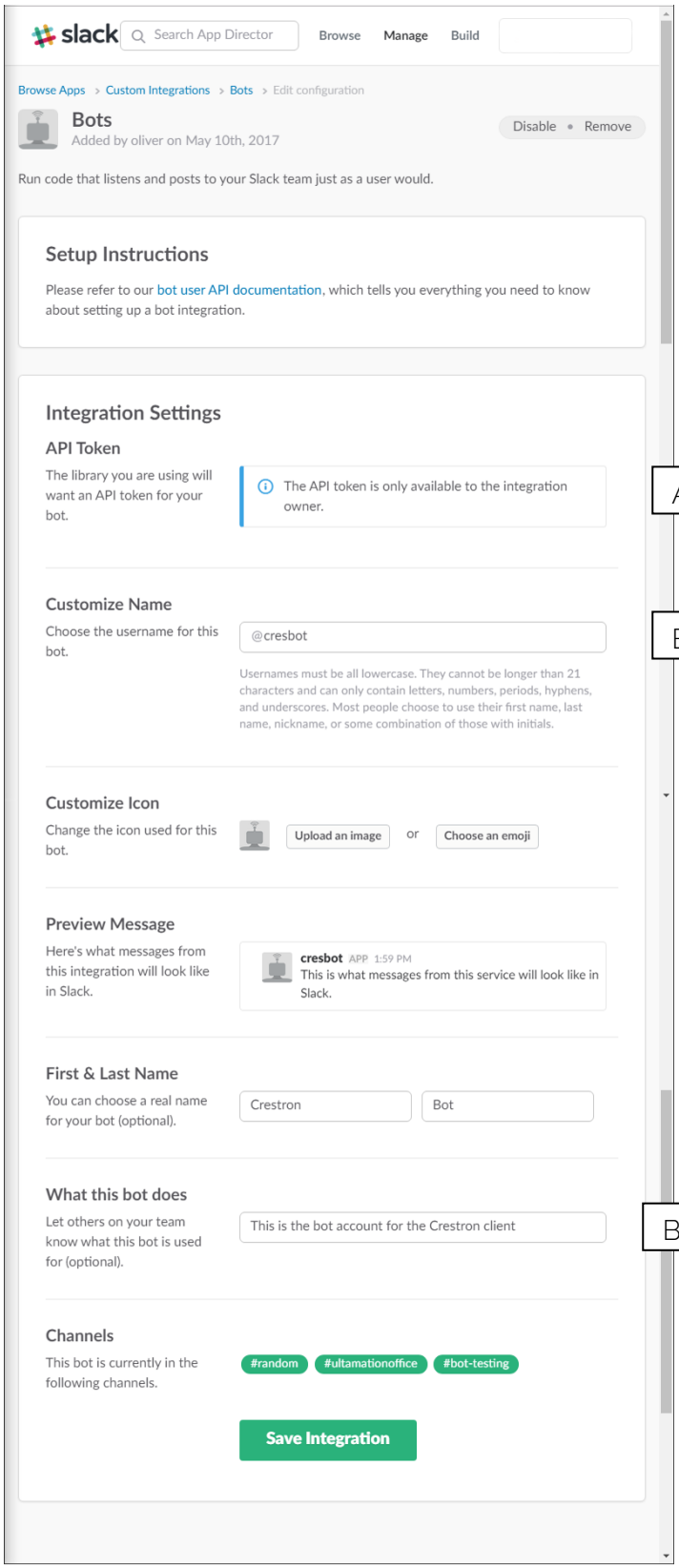
The licence key is tied to the processor MAC address and can be used for multiple instances of the module on the same processor. Therefore multiple Slack Client modules can be used in the same project and you will only require one licence.

AccountToken – The Slack token is used to connect to the slack bot and to authenticate the module with the Slack Bot.

To create an account token, go to the url: https://(YourSlackAccountName).slack.com/apps/manage/custom-integrations/.

On this page, under the title Custom Integrations, select "Bots" to start the set up process and select "Add Configuration". On the following page, enter a name for your bot (this will be used when displaying any messages from the module) and select "Add bot integration". The next page displays information on the bot including the account token shown under the heading API Token. Copy and paste the token into the module.

The bot then needs to be invited to the appropriate slack channels. To do this, within Slack open a channel and post this message substituting with the Bot Name "/invite@ (BotName)". The bot will now be added to that slack channel. Repeat for any other channels.



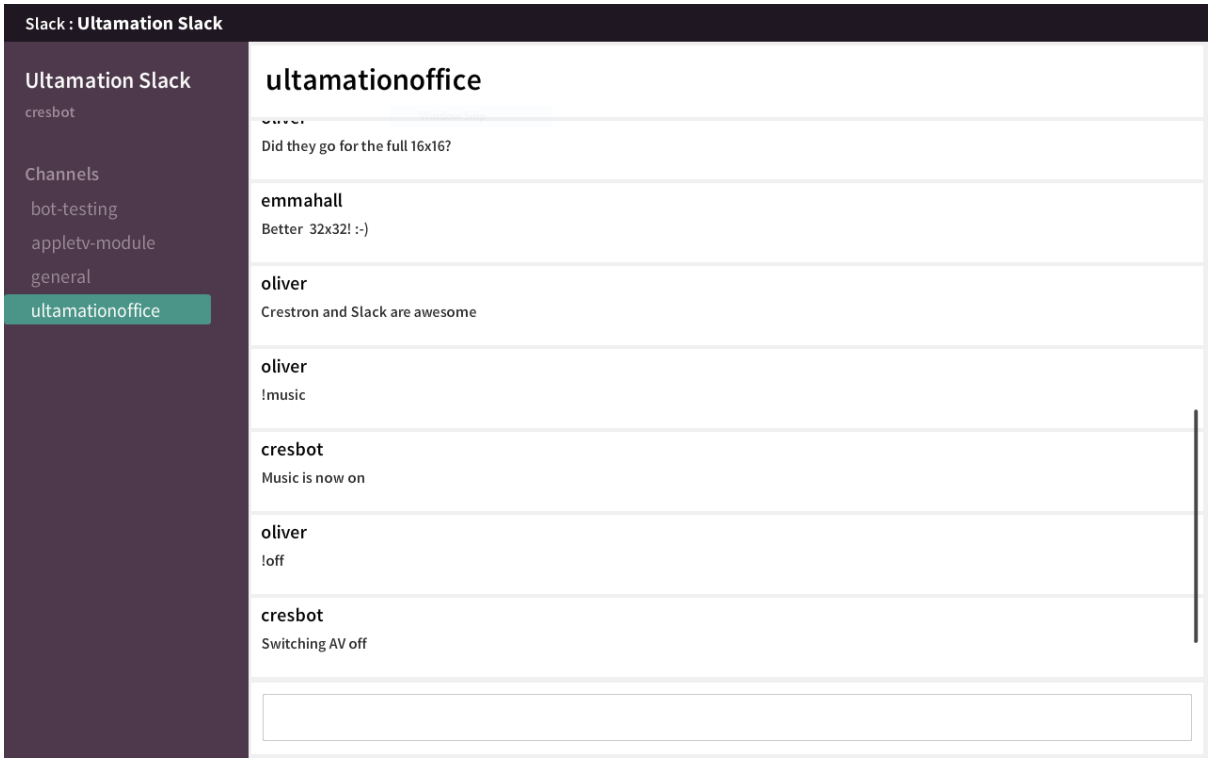
Account Token displayed here

Bot Name

Bot Description

MessageHistoryLimit –The amount of messages the module will pull from the Slack channel upon connection.

LicenceKey – The module will function, without any restrictions, for 1 hour without a licence key so that integrators can "try before you buy". For continued use, the module requires a licence key that is generated at the time of purchase from the Ultamation' Store and is linked, at that time, to the information provided for the processor MAC address. The licence key will be delivered via email to the address linked to the account used at checkout.



Example touch panel included with the module.

PROGRAMMING NOTES

Each of the module files (see below) should be placed either in the host program's project folder, or to make the Slack Client module available to all Crestron programs, in the SIMPL Windows installation's User SIMPL+ (for .usp, .ush and .clz files) directories. This pdf should be placed in the same directory for SIMPL's F1 help function to work properly.

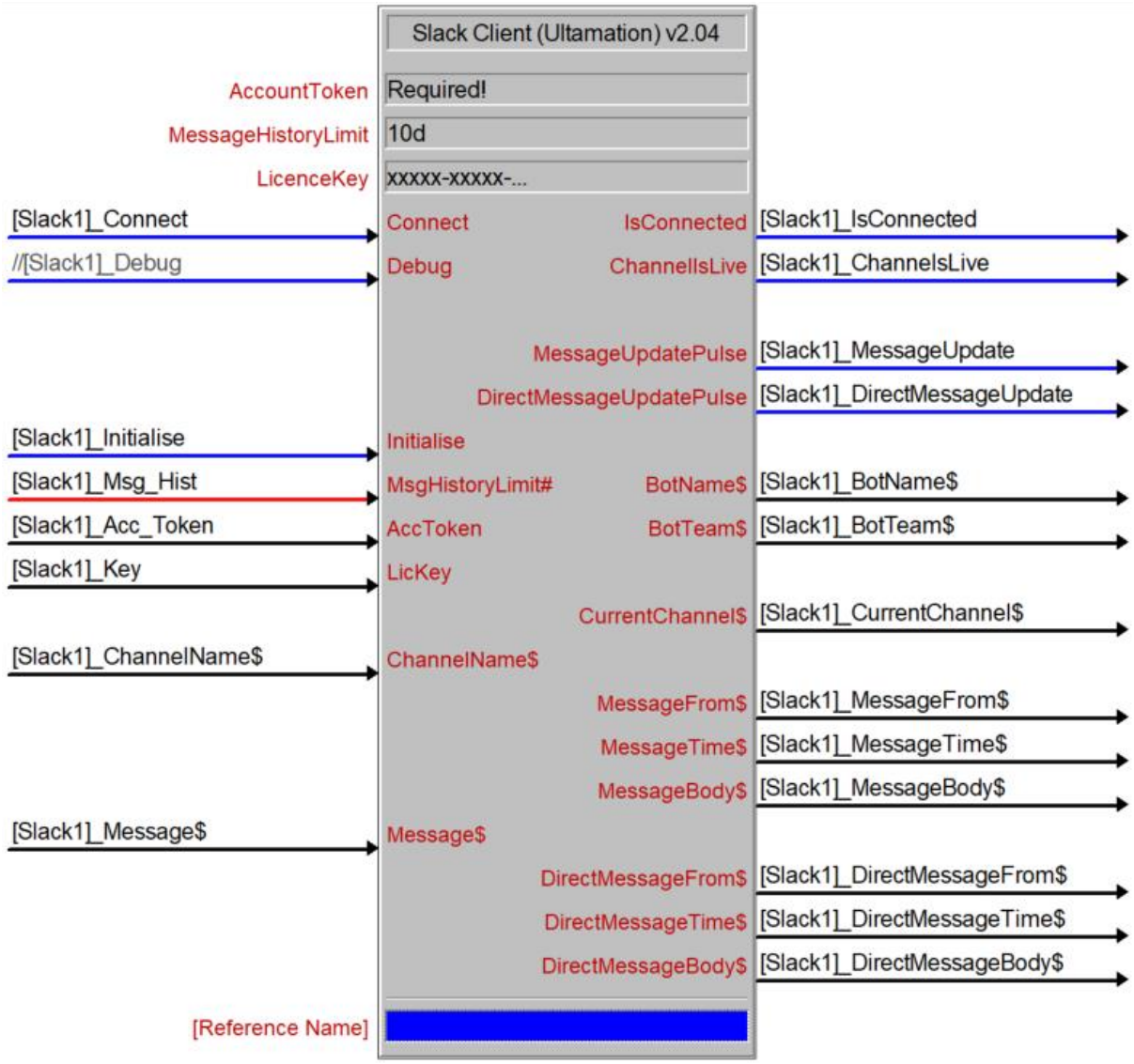
The module consists of a single SIMPL+ wrapper module to the core SIMPL# driver:

- Slack Client (Ultamation) v2.04.usp & .ush
- SlackClient2.clz
- Slack Client Datasheet (Ultamation) v2.04.pdf (this file)

Multiple instances of the Slack module can be inserted into a program, to display more than one channel at one time. Or the channel name serial can be changed to select a different Slack channel.

Setup Notes:

- As of v2.04 - to fully connect the module you must set the Connect signal high, then pulse the Initialise signal.
- As requested - values for LicenceKey, AccountToken, and MessageHistoryLimit can now be given via either parameters or signals.



MessageHistoryLimit – The maximum number of messages you want to the module to retrieve from a channel's history **(Can be provided as a parameter or via the signal)**

AccountToken – The token required to access your Slack account **(Can be provided as a parameter or via the signal)**

LicenceKey – The licence key from Ultamation. Without a key the module will work for 1 hour. **(Can be provided as a parameter or via the signal)**

Connect **–** When this signal is high the module will prepare to connect to the Slack account. To fully connect the Initialise signal must also be pulsed

Debug – Set high to enter debug mode. This prints extra debug information to the console output that can be sent to Ultamation if you are having issues with the module.

Initialise – Must be pulsed to complete the initialisation of the module

IsConnected – This signal will go high when the module successfully connects to a Slack channel.

ChannelsLive – This signal will go high when the slack module has connected to slack and finished gathering the message history.

MessageUpdatePulse **–** This signal is pulsed when a message is received.

DirectMessageUpdatePulse **–** This signal is pulsed when a direct message is received.

BotTeam$ **–** This serial signal will hold the Slack team name.

ChannelName$ **–** The name of the Slack channel the module will connect to. The Slack Bot must have joined the channel first in order to receive any messages.

CurrentChannel$ **-** This serial signal holds the currently connected Slack channel

MessageFrom$ **–** This serial signal holds the last message sent, sender's name, from the currently selected Slack channel.

Message$ **–** Pass a serial through this signal to send a message to the currently connected Slack channel.

MessageBody$ **–**This serial signal will hold the last message sent on the currently selected Slack channel.

DirectMessageFrom$ **–** This serial signal holds the last direct message sent to the module

DirectMessageBody$ **–** This serial signal holds the last direct messages, sent to this module, sender's name.

Please note: only the information noted above is supported by the module.

Licence

This module (including software, images and any and all other associated assets distributed as part of the purchased download package) is licenced on a PER PROCESSOR basis.

A licence key is generated at the point of purchase and is linked at that time to specific information that MUST be provided at the time of purchase. A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

The licence key for each processor will be delivered via email along with links to download the module. There is no physical delivery.

The module is provided without any warranty with respect to Slack's API. We will endeavour, through best efforts, to maintain the module's functionality and any bug fixes will be provided free-of-charge. Additional functionality may be released as a variation of this module and this will be a separate, purchasable, product.

Changelog

1.01

- Updated API calls to be compatible with the Slack API update

2.01

- Updated to be database 200 compatible

- Updated command strings to match the API changes

2.02

- Fixed message retrieval limit being sent incorrectly (PullHistory parameter)

- Changed PullHistory to MessageHistoryLimit for clarification, and added info to docs

2.03

- Added the option of signal versions of the parameters – message history, licence key, and account token.

2.04

- Added a Debug signal to enter Debug mode