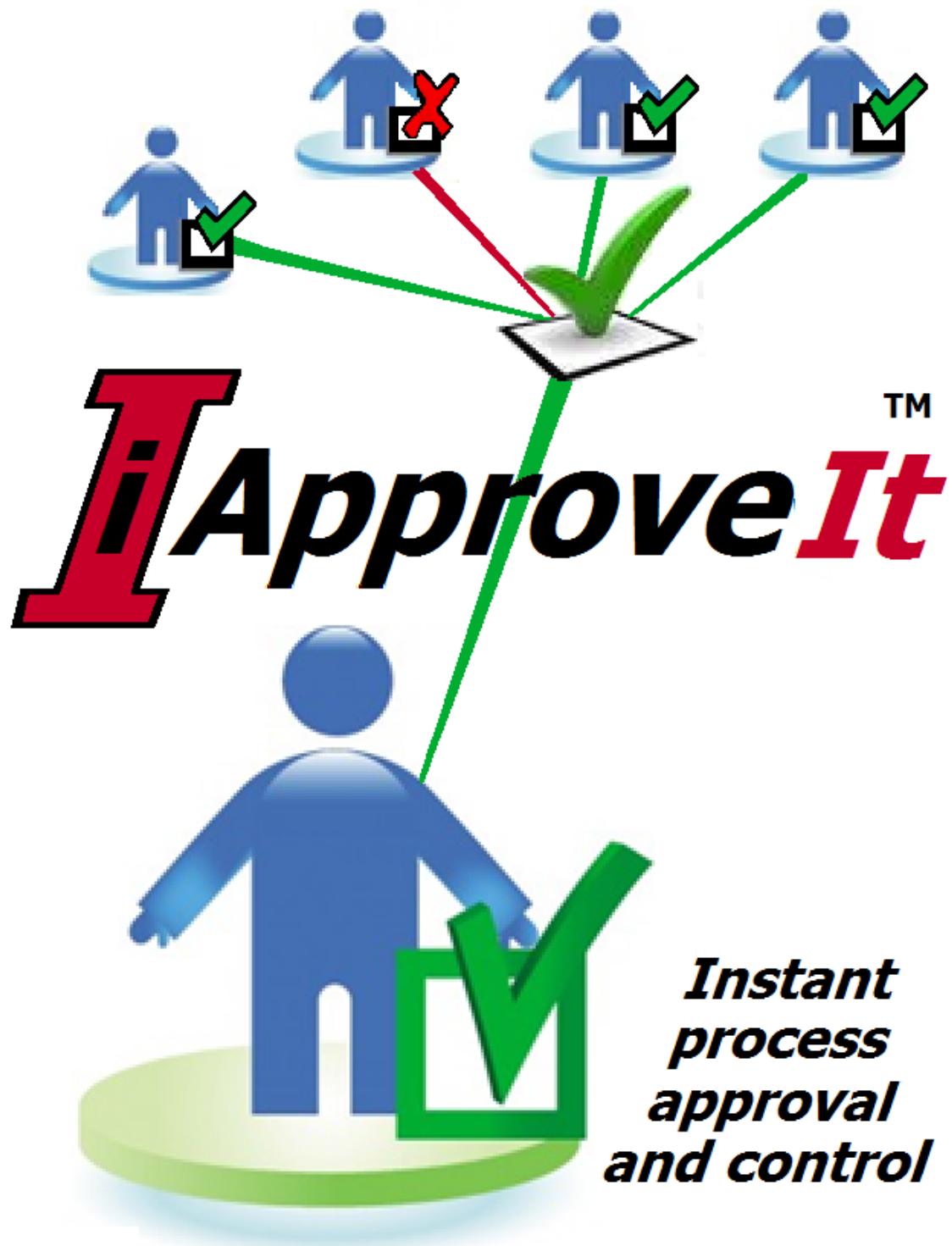


# iApproveIt Client Version 1.0

## User Guide





|  |           |
|--|-----------|
| <b>Product Delineation .....</b>                     | <b>4</b>  |
| <b>iApproveIt Overview.....</b>                      | <b>5</b>  |
| <b>Customer Support .....</b>                        | <b>9</b>  |
| <b>Supported Platforms .....</b>                     | <b>9</b>  |
| <b>Installing the iApproveIt Client.....</b>         | <b>10</b> |
| <b>Software Subscription .....</b>                   | <b>10</b> |
| <b>iApproveIt Web Client.....</b>                    | <b>11</b> |
| <b>iApproveIt Registration.....</b>                  | <b>11</b> |
| <b>iApproveIt Associates .....</b>                   | <b>12</b> |
| Inviting an Associate.....                           | 13        |
| Accepting/Rejecting Associate Invites .....          | 14        |
| Managing Current Associates .....                    | 15        |
| <b>iApproveIt Projects.....</b>                      | <b>16</b> |
| Create Projects .....                                | 16        |
| Edit “Self-Owned” Projects .....                     | 17        |
| Edit “Associated-Owned” Projects .....               | 18        |
| Using and Testing Projects from the Web Client ..... | 19        |
| <b>Managing Your Configuration .....</b>             | <b>24</b> |
| Show Your Profile .....                              | 24        |
| Edit Your Profile .....                              | 25        |
| Edit Your Profile: Verify Your Mobile Number .....   | 26        |
| Edit Your Profile: Verify Your Webhook Channels..... | 27        |
| Documentation and Download.....                      | 28        |
| Upgrade Your Subscription .....                      | 28        |
| <b>iApproveIt Dashboard.....</b>                     | <b>28</b> |
| Pending Unanswered Responses.....                    | 30        |
| Pending Answered Reponses .....                      | 31        |
| Recently Expired/Settled Reponses .....              | 32        |
| <b>iApproveIt (command line).....</b>                | <b>33</b> |
| <b>iApproveIt dialog_types.....</b>                  | <b>35</b> |
| <b>Examples: Using the iApproveIt Client.....</b>    | <b>37</b> |
| <b>“Permission” or “Voting” examples.....</b>        | <b>38</b> |
| approve_any - examples.....                          | 39        |
| approve_majority - examples.....                     | 41        |
| approve_unanimous - examples.....                    | 43        |
| first_reply - examples .....                         | 44        |
| objection - examples .....                           | 46        |
| <b>“Advice” examples.....</b>                        | <b>48</b> |
| advice - examples.....                               | 48        |



|   |           |
|---|-----------|
| <b>“Notification Only” examples .....</b>   | <b>50</b> |
| gripe - example .....                       | 51        |
| tell – examples .....                       | 52        |
| silent - example .....                      | 53        |
| Misc. or “Multi-call” examples.....         | 54        |
| “Hierarchical Policy” - example .....       | 55        |
| “Other Product Integration” - examples..... | 56        |



## Product Delineation

The phrase “**iApproveIt Web Client**” is used throughout this document to mean the iApproveIt Web Interface that is accessed via a desktop or mobile web browser. The web interface is available here:

<https://iapproveit.abs-consulting.com/iApproveIt>

The phrase “**iApproveIt Client**” is used throughout this document to mean the iApproveIt executable that is used to control policy which resides on Apple MacOS, Microsoft Windows, and various Unix and Linux machines. The **iApproveIt Client** is available for download *free* of charge from within the **iApproveIt Web Client**.

The phrase “**iApproveIt API**” is used throughout this document to mean the iApproveIt API Server owned by ABS that is responsible for sending notifications and storing User and Project data. The Server is accessed by the “Client” software; the end-user need not be concerned with the server except to gain conceptual or architectural knowledge.

The phrase “**iApproveIt**” (without the words “client”, “device” or “server”) is generally meant to refer to the iApproveIt product as a whole.



## iApproveIt Overview

**iApproveIt** is an application that allows for the real-time interactive approval and notification of watched or controlled events. The insertion of such control was previously cumbersome because it was performed via email which required the email receiver to respond before the process could continue which resulted in delays if the receiver was not currently at their desk to read and respond to the email. Furthermore, the receiver (or approver) could not guarantee that their approval or disapproval of the action was adhered to as emails can be lost, misinterpreted, or even ignored.

Though text messages provide usually faster responses, again the approver has no guarantee that their wishes are adhered to by the requesting user. Also, text-only base responses required that the requestor had the approvers mobile number which is not always the desired case. With iApproveIt, notifications can be sent to an approver's phone or associated channel without having to know the approver's mobile phone number or channel link.

With iApproveIt, the “approval process” is imbedded in your process so proper controls are provided since the responses provide the control. iApproveIt process approval and/or notification is in real-time thus lowering end-to-end process time from minutes or hours to just seconds.

Users are placed in **Projects** (or groups of individuals). Projects are created and intended to be notified or queried together when certain actions are performed or about to be performed.

For example, the Project “**Company CXOs**” might contain an organization's C-Level executives (CEO, CTO, CFO, CSO, etc.). Now the “Company CXO's” can be notified when certain actions are performed or even *queried* for their approval before the action is performed. The “Company CXOs” can instantly vote on whether and action should be allowed or disallowed depending on the resulting desired condition. For example, if a controlled action was attempted by a user that required CXO “approval” the action could be allowed or not pending the instant CXO

Group opinion of the CXO members where either:

- First responding CXO response determines the outcome
- Any CXO member “approval” allows the action
- Any CXO “objection” prohibits the action
- A majority CXO opinion determines the outcome
- A “unanimous” CXO approval is required to allow the action



Additionally, Project “**Company CXOs**” might wish to be notified is before or after allowed actions are performed or denied by company policy. For example, the CXO members might want to be notified when someone starts or completes a particular program or program task or even when users attempt an action or be prevented from starting a program or task. For example, if a controlled action was attempted by a user that required CXO “notification” the action could be allowed or denied and the CXOs would either:

- The controlled program or task is allowed and All CXO members are notified of the action.
- The controlled program or task is disallowed and All CXO members are notified of the attempt

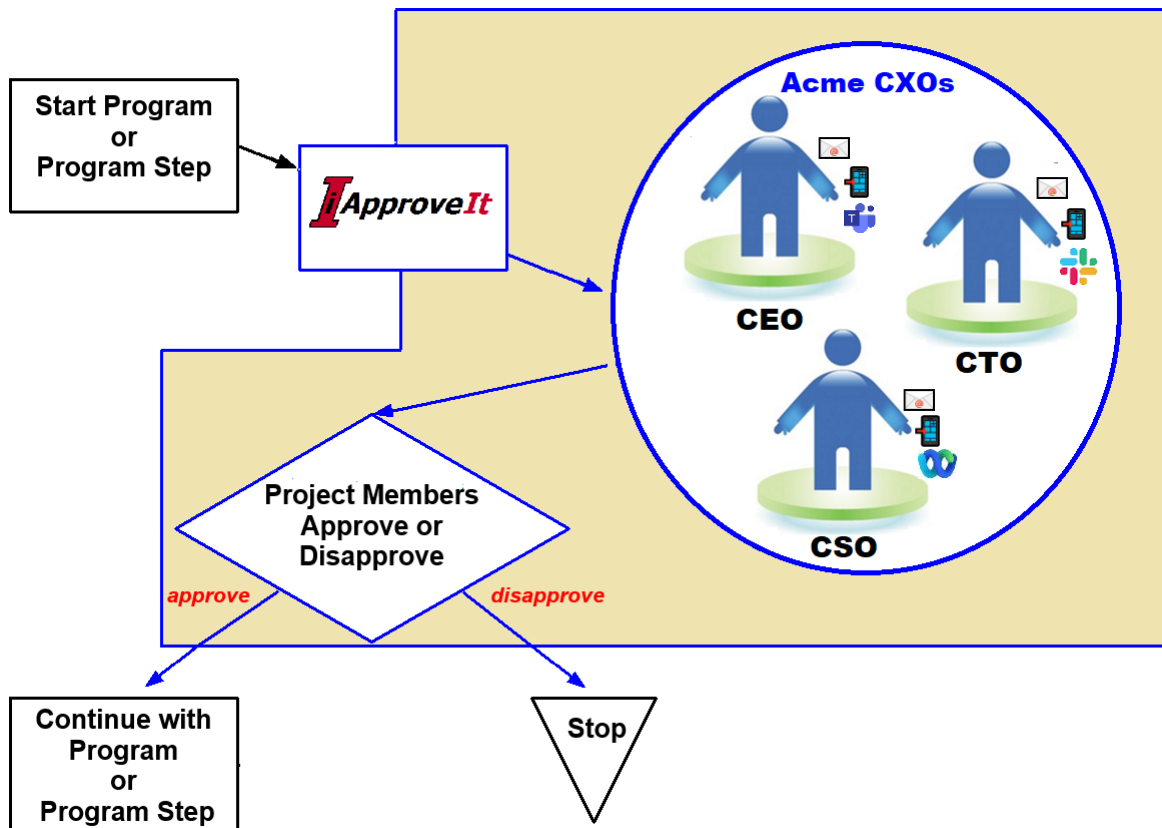


Project Members can elect to be contacted simultaneously through many devices or media like **email**, SMS on their **mobile phone**, or through application channels through webhooks like **Microsoft Teams**, **Slack**, **RingCentral**, **WebEx** and more.



The **iApproveIt Client** program (or iApproveIt API call) establishes communication between the process that is being controlled or audited and the devices of **Project Members**.

The **Project** can be used for any number of actions that the organization wishes to control or provide notification for; each action can be controlled differently using different voting parameters. **Users** can create many Projects, be a **Member** of many Projects and Projects can have many Members.



The individuals that perform actions do not need any software or software license, only those that wish to administer Projects or be notified or queried for their approval or opinions before or after any controlled action is performed.



For example, when a user (Tony) attempts to perform some controlled action (in this case start the QA Build). Tony is notified that what they are attempting requires approval from Bruce and Clark in real-time they must wait 1 minute (a configurable number of minutes for each request) for their **unanimous** approval to do so before Tony's build attempt can proceed.

```
c:\build_scripts\QA_build.bat

First asking these people:

- Bruce Wayne
- Clark Kent
- Bruce Banner

for their real-time approval to perform this action...

- Proceed with the QA Build?

waiting up to 2 minutes for the UNANIMOUS agreement of ALL of
these people to decide if you can perform this action...
```

The members in the “QA Build Team” Project (in this case Clark and Bruce) are notified either via email and/or SMS and asked for permission to perform this task. This messaged/response ability is available if they are in a meeting, at their desk, at lunch or on vacation (if you are bold); try getting that dynamic a response from email.

In this case permission is granted only if all Members provide their ‘Unanimous’ approval. Project Members select a link in their SMS text on their phone (or email) and are shown that a real-time request is awaiting their approval or advise and how long they have to respond on their phone or desktop browser like the image to the right. Then can Approve or Reject the action and even provide some short comment that Tony can see as well in real-time.

Hello **Clark**,

Please be advised, **Tony Stark** is performing an action:

**"Proceed with the QA Build?"**

This message is of the **approve\_unanimous** type so, your immediate real-time permission/advise is requested for this action as one of the **3** members of the **"QA Team" Project**.

The **unanimous** acceptance of all of the Project members is required for this action to be allowed for Tony.

Comment:

(optional)

(Time Remaining to Respond = 28 seconds)

I Approve

I Reject

Only when all Members provide their approval does the program proceed as depicted here:

```
c:\build_scripts\QA_build.bat

First asking these people:

- Bruce Wayne
- Clark Kent
- Bruce Banner

for their real-time approval to perform this action...

- Proceed with the QA Build?

waiting up to 2 minutes for the UNANIMOUS agreement of ALL of these people to decide if you can perform
this action...
=====
- Bruce Wayne APPROVES
- Clark Kent APPROVES and provided a comment of:
  Up, up and away
- Bruce Banner APPROVES and provided a comment of:
  Hulk wants to SMASH
=====
3 of the 3 members APPROVED the action.
=====
Resuming with this action.
=====
```



If *any* Member replied with “**Disapprove**” then the action would have been aborted as depicted below:

```
c:\build_scripts\QA_build.bat

First asking these people:

- Bruce Wayne
- Clark Kent
- Bruce Banner

for their real-time approval to perform this action...

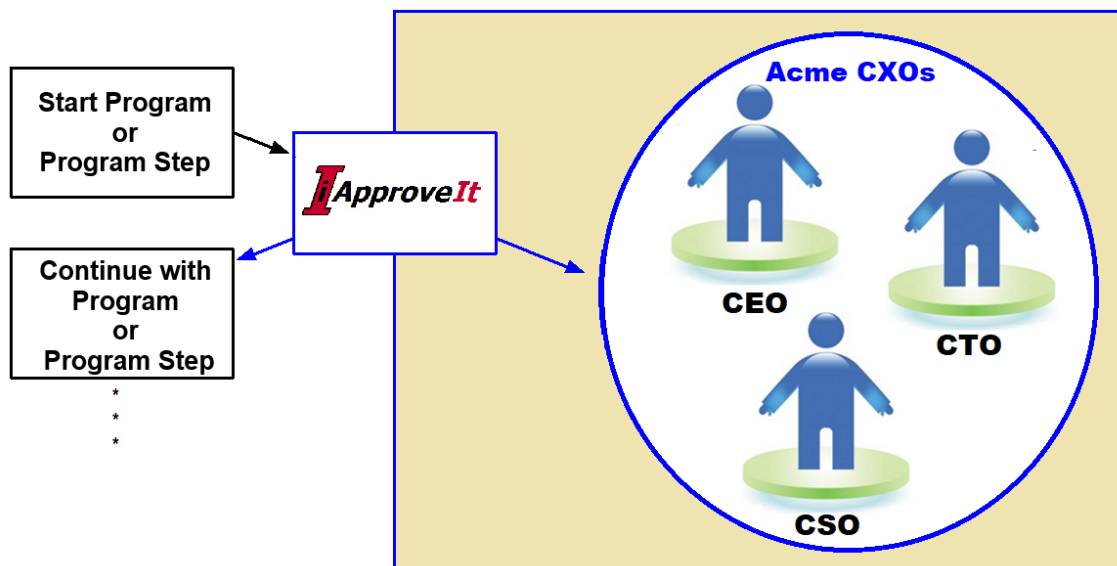
- Proceed with the QA Build?

waiting up to 2 minutes for the UNANIMOUS agreement of ALL of these people to decide if you can perform
this action...
=====
- Clark Kent APPROVES
- Bruce Wayne APPROVES and provided a comment of:
  Be careful, it is dangerous.
- Bruce Banner DISAPPROVES and provided a comment of:
  Hulk wants to SMASH, Hulk always wants to SMASH
=====
Only 2 of the 3 members APPROVED the action.
=====
Prohibiting this action.
=====
```

As you can see not only can the Members *direct* policy with “**approvals**” or “**disapprovals**”, but they can even provide advice or comments back to the user. Again, you just can’t do this with email or texting.

You can now create a process that includes those Managers that want control where before you might not because they could not respond fast enough to include their approval in real-time thus making the process to slow to proceed.

The **iApproveIt Client** can also be used for real-time notification when that is all that is needed. Insert a call to iApproveIt and instantly inform Project Members of what is happening directly to their email, mobile phones or webhook channels. So, you can keep those bosses, system administrators or other peers in the know.







## Customer Support

To obtain additional information on iApproveIt or other products or services offered by A Better Solution, Inc., visit our web site at [www.abs-consulting.com](http://www.abs-consulting.com). To report problems with the iApproveIt software or documentation, please send e-mail to: [iApproveIt\\_support@abs-consulting.com](mailto:iApproveIt_support@abs-consulting.com).

## Supported Platforms

The iApproveIt Client can run on various platforms and Operating Systems. The following operating systems are supported for iApproveIt Client:

❖ **UNIX:**

- **AIX**
- **CentOS**
- **HP-UX**
- **RedHat Linux**
- **Sun/Solaris**
- **SUSE Linux**
- **Ubuntu Linux**

❖ **WINDOWS:**

- **Windows 9x**
- **Windows NT**
- **Windows 2000 and higher**

❖ **MAC**

- **MacOS**



## Installing the iApproveIt Client

The **iApproveIt Client** is available for download from within the **iApproveIt Web Client** or an email link can be sent. There is **no cost** for this client software. Once you download the appropriate zip file, just extract the single client binary in a common location where it would need to be executed.

The screenshot shows the iApproveIt web client interface. At the top is the iApproveIt logo. Below it is a navigation bar with tabs: Associates, Projects (selected), Configure, and Logout. Under the Projects tab, there are sub-tabs: Show, Edit, Docs, and Upgrade. A green-bordered box contains the text "Download Client Binaries or Documentation here...". Below this, there are links for "iApproveIt Documentation" (with a PDF icon) and "iApproveIt Web and Client vCard" (with a vCard icon). The client version is listed as "iApproveIt Client Version: 1.0" and the build is "Build: 20210831.12:20p". A table lists supported platforms and download links. To the right of the table is an email icon. At the bottom, a green bar contains the copyright notice: "iApproveIt ©2021 A Better Solution, Inc. All rights reserved."

| Supported Platforms | Download Links (size)                       | Send Links Via Email |
|---------------------|---|----------------------|
| Windows             | <a href="#">iApproveIt_win.zip</a> (193 KB) |                      |
| RedHat / CentOS     | <a href="#">iApproveIt_rh.zip</a> (19 KB)   |                      |
| Ubuntu              | <a href="#">iApproveIt_ub.zip</a> (20 KB)   |                      |
| openSUSE            | <a href="#">iApproveIt_suse.zip</a> (20 KB) |                      |
| Sun / Solaris       | <a href="#">iApproveIt_sun.zip</a> (21 KB)  |                      |
| HP-UX               | <a href="#">iApproveIt_hp.Z</a> (42 KB)     |                      |
| AIX                 | <a href="#">iApproveIt_aix.Z</a> (64 KB)    |                      |
| MacOS               | <a href="#">iApproveIt_mac.zip</a> (18 KB)  |                      |

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.

## Software Subscription

**iApproveIt is provided as a subscription service.** There are three (3) levels of service; **Silver**, **Gold** and **Platinum**. The silver level is free and fully functional with all features fully enabled. At the Silver level you are just limited as to the number of Projects and Number of Project Members. Subsequent levels allow for more Projects and Project Members.

You can use the “client” to embed real-time notification and approval actions in your new or existing processes by calling the **iApproveIt Client** program (or the soon-to-be public API) with an appropriate **Project\_ID** and **Passphrase**.



## iApproveIt Web Client

The **iApproveIt Web Client** allows access to iApproveIt application via a web browser either on the desktop, tablet or mobile devices. From here users can manage projects, profiles, associates and subscriptions.

### iApproveIt Registration

To register for an account with iApproveIt, open a web browser to:

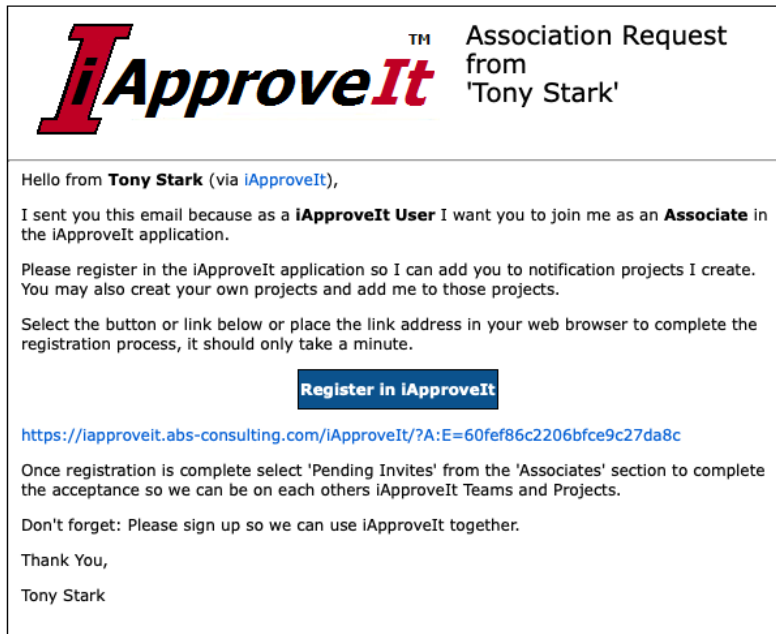
<https://iapproveit.abs-consulting.com/iApproveIt/>

or you might get to the register page by following a link you receive in an **Invite to iApproveIt** email you might receive from another user you know.

Click the **Register** menu item and complete the required **User Profile Information** fields.

Then select the **Register for iApproveIt** button which will send an email-verification email to the email address you supplied to verify your account.

Here is an example of the registration email.



Please enter your **required** First Name.

First Name:  ✓

Last Name:  ✓

Email:  ✓

Logon ID:  ✓

Password:  ✓

Password:  ✓

**Register for iApproveIt**

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.

Select the **“Register in iApproveIt”** button or the link in the email to confirm your email address, activate your account and take you to the **Logon** screen.

**iApproveIt**

**Register**

Please enter the **required** Logon ID in order to logon.

Logon ID:

Password:

**Logon to iApproveIt**

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.



## ***iApproveIt Associates***

As an iApproveIt user you create associations with other users of iApproveIt. This allows you to add ***Associates*** to your projects and be added to projects created by other ***Associates***. There is no limit to the number of Associates you can add in any subscription-level of iApproveIt. Freely invite anyone you expect that will be involved in decision making for any of the controlled programs or processes you expect to create. Examples of people you might expect to invite are:

- Any of your managers or other higher in the organization that might need to approve actions for you or other members of your development groups etc.
- Any of that might be responsible for making or controlling process for others in their part of the organization (CM Leads, QA Leads, HR Managers System Administrators, etc.).

Those that make policy should be invited. Those that follow policy do not need to be added, but they could be as they could use iApproveIt to create policy for others or even themselves as they might want notification for certain actions as well. You are not limited on the number of Associates you can have.



## Inviting an Associate

After logging into iApproveIt you would navigate to the *Associates Screen* and choose the **Invite** menu item.

From here complete the form by adding the user's email address and a comment to include in the invite email.

If the invited user is new to iApproveIt, they will receive an email from you inviting them to *Sign-Up* (register in iApproveIt) and then *Accept* your **Associate Invite Request**.

Existing users will receive an email to just *Accept* your **Associate Invite Request** and will also be notified when logging on the application that they have a pending Invite that requires their attention.

The **invitation email** is sent to the user with instructions on how to register in iApproveIt and how to accept your invite, but you can input an **Optional Comment** that is sent with the email if you feel it is necessary. If an Optional Comment is sent it is prefaced with “**Don't Forget:**” as depicted below:



## Accepting/Rejecting Associate Invites

Anytime you logon to the Web Interface you are notified if you have any **Pending Invites** received from others that you may wish to address. You can select the **Associate Invite** link on the **Welcome** screen or the **Pending Invites** menu item in the **Associates** Sub-menu to get to the **Pending Invites** screen.

Hello Bruce (batman),

Have fun using the iApproveIt Application ...

- You currently have **1 Received Associate Invite** awaiting your attention.
- You currently do not have a mobile phone registered in the application. Doing so allows for faster notifications making sure **your input** is part of the decision-making process. Your phone number is not shared with anyone, not even your Team Members.

[Register your phone](#) now, for faster approvals and decisions.

Here you see a list of your **pending invites** awaiting your attention.

You can choose to individually **accept/reject the selected invite** or **accept/reject all pending invites**.

**iApproveIt™**

Associates Projects Configure Logout

Invite Pending Invites List/Edit

List, Add or Revoke or Reject Pending **Associate Invites**.

You currently have no open 'Associate Invites'.

[Invite](#) others to be your Associates.

You received **1** pending Associate Invite(s) from:

Associate Name (Logon ID)  
Tony Stark (ironman) ▾

Send RSVP Email Notification(s): ☒

Accept Selected Invite Reject Selected Invite

Accept ALL Invites Reject ALL Invites

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.

You also have the option of notifying (or not notifying) the invite sender of your decision by toggling the **Send RSVP Email Notification** checkbox.

Send RSVP Email Notification(s): ☒



## Managing Current Associates

The **List/Edit** screen allows you to manage your Associates.

### A pulldown list of your Associates

(if any) and can select one to view that Associate's membership in your **Projects** and your membership in their Projects (if any).

If the **Selected Associate** is a member of any of your Projects, a **list of those Projects** and you are given the option to remove:

- the Associate from *the selected* Project
- the Associate from *all* of your Projects
- the Associate from *your list* of Associates.

You can choose to send (or not send) an email to the Selected Associated when removing them from your Project(s) by toggling the **Send remove Email Notification** checkbox.

**Send Removal Email Notification(s): ☒**

If you are a member of any of a **Selected Associate's Projects**, you would be shown the number of such Projects and provided a **link** that would take you to the **Edit** menu item in the **Projects** Sub-menu to get to the **Projects Edit** screen for the first of Associate's Projects of which you are a Member. It is there that you would remove *your* membership in *their* project if you desired.

**You ARE a member of 1 of ironman's Projects.**  
You can view those Projects [here](#)





## iApproveIt Projects

**Projects** (or groups of individuals) are created and intended to be notified or queried together as a group when certain actions are performed or about to be performed. From the Projects Tab you will create, manage, and sometimes use projects (mainly for Testing or Associate Polling).

### Create Projects

To create a project, complete the form provided on the **Projects:Create** screen. All the fields are required and guidance on valid values is given. Description of each required field follow:

- **Project Name** – A unique name within your owned Projects that you and its members or perspective members can use to identify the Project. This Project Name show in user visible selection list.
- **Description** – A short description on the Project. This is only seen by the Project Owner.
- **Passphrase** – Is something you might share with others that you might allow to create request instances of this Project (those individuals that might be responsible for creating policy).
- **Min. and Max Duration** – These define for the project the *minimum* and *maximum* values duration for which project instance requests of the project can be defined. In this example, instance requests can have a **maximum wait time** between 1 and 10 minutes inclusive defined for any request of this Project Type.
- **Privacy** – This controls who from your associates can see your Projects. For **Private** Projects, only members of the Project may see the Project Name in Project List. For **Public** Projects can be seen by all of your associates. Only your Associates can see your Projects. Public Project are prefaced with the **Public Project Indicator** 🔍 in Project List or Pulldowns while Private Projects are prefaced with the **Private Project Indicator** 🔑.

**iApproveIt™**

Associates Projects Configure Logout

Create Edit Use

Please enter the **required Project Name**. This can contain 3-18 characters within the [a..z], [A..Z], [0..9], [!+=-\_.'] character sets.

You currently have **13** Projects defined:

- C\_Level
- Documentation Team
- Lead Developers
- Project Managers

**New Project Details**

Project Name: Super Heros ✓

Description: Gadget heros ✓

Passphrase: GetItDone ✓

Min. Duration: 1 minute ✓

Max. Duration: 10 minutes ✓

Privacy: Public ☒ Private ☐

Create New Project Clear Project

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.

Selecting an existing Project in the **Project List** will prepopulate all field with the current values of that Project as a quick means to clone a project from an existing project.

Populate the required fields then select **Create New Project** button to create a new Project.







## Edit “Self-Owned” Projects

To edit an existing Project that you own/created, navigate to the **Projects:Edit** screen as depicted to the right.

Select from the “*Self-Owned*” **Projects List**. Projects are listed by Project Name in alphabetical order.



Public Project are prefaced with the **Public Project Indicator**  in Project List or Pulldowns while Private Projects are prefaced with the **Private Project Indicator** .

When a “*Self-Owned*” **Project** is selected, the Project attributes **Name**, **Description**,


**Passphrase**, **Min. Duration**, **Max. Duration** and **Privacy** are displayed and editable.


You can also add/remove Project Membership to include/remove yourself or any of your Associates by:

- Selecting the **Apply All** ☒ to add yourself and your Associates to the Project.
- Selecting the **Apply All** ☐ to remove yourself and your Associates to the Project.
- Individually selecting ☒ or ☐ for yourself or an Associated to toggle that users Membership.

In the **Associates List**, Associates are prefaced with the **Email-Only Indicator**  or the **SMS-Enabled Indicator** .

When **Project Attributes** and **Membership** are to your liking, select the **Save Project** button to save the data.

This  widget provides a shortcut to the **Project:Use** screen with the selected Project Name.

This  widget opens a **Project Usage Dialog** that displays how old the Project is and how frequently it has been used.



## Edit “Associated-Owned” Projects

To edit an existing Project that is owned by one of your Associated, navigate to the **Projects:Edit** screen as depicted to the right.

Select from the “*Associated-Owned*” **Projects List**. You can see the Public Projects owned by your Associates or Private Projects if you are currently a Member of the Private Project.

Projects are listed by Project Owner then Project Name in alphabetical order

Select from your 15 “Associate-Owned” Project(s):

★|☆ (Project Owner) Project Name

- ★ (cciug) first project
- ★ (jayb72) scm\_supprt
- ★ (jjjjjjjj) Sara Project
- ★ (johncollins) Project JC
- ★ (kriscochran) Test
- ★ (kwcreager) Kens First Project
- ★ (rjcarter) A Projdd
- ★ (ricarter) a vacm

Projects that you are currently a member of are prefaced with the **Project Member Indicator** ★, while Project List Pulldown while Projects where you are NOT a member are prefaced with the **Project Non-Member Indicator** ☆.

Once an “Associated-Owned” Project is selected, your Project Membership is displayed, and you can toggle you Membership to the Project.

You will see either see indicating you are currently NOT a Project Member. In which case select the **Add me to this Project** button to add yourself to the Project.

You **ARE NOT** a Project member

Add me to this Project

Or you will see indicating you **ARE** currently a Project Member. In which case select the **Remove me from this Project** button to remove yourself to the Project. With either action the Project Owner is automatically notified of this change by email.

You **ARE** a Project member

Remove me from this Project



## Using and Testing Projects from the Web Client

The **Projects:Use** screen allows you to test and use *instances of your Projects* in limited fashion. This screen is excellent for testing your Projects to get a feel of how it will appear to your users during client calls or for executing real-time *Associate Polls and request for Advice*. In most instances Project request will come from use of the [iApproveIt Client](#).

First, if you have a single Project prior to entering this screen, the screen is pre-populated with that Project's information. If you had more than one Project, then you are first shown a **Project Selector** like this one where you could select the Project you wish to display.

Select to view from (2) "Self-Owned" Project(s)

(privacy) Project Name

▼

🔍 Marvel

🔑 QA Team

The **Project Names** are prefaced with the *Public Project Indicator* 🔍 or the *Private Project Indicator* 🔑.

The **"Self-Owned" Project Details** section shows summary information about your selected Project - the **Project Name**, **Description** and the current **Project Members** are displayed.

The **Project Members** are prefaced with the *Email-Only Indicator* 📧 or the *SMS-Enabled Indicator* 📱.

If you need to make changes to the Project, click on the **Project Editor Shortcut icon** ✎ which will take you directly to the [Project:Edit](#) screen with the current Project pre-selected for display.

The **Build Project Request** section allows you to build the *Client Request Command* that can be used to request Real-Time input from the **Project Members** to be used as a *process point* in external program or process or for *Real-Time Web Request*.

### Build Project Request

Request Kind: advice ▼

Max. Duration: 1 minute ▼

Comment: (required)



Select a **Request Kind** that defines how the request will act during web-side or client-side execution. For all actions, all members of the Project are informed of the action, the Request Kind determines how or if the Members input is interpreted determining how that input will change the process flow for the effected end-user. Descriptions of the different Request Kinds are in the table below.

| Request Kind             | Purpose  |
|--------------------------|--|
| <b>advice</b>            | Inform the members of the Project and allow them to provide real-time advice to the user attempting the controlled for up to <b>Max. Duration Minutes</b> , then allow the action to proceed.    |
| <b>approve_any</b>       | Request approval from the Project members and approve the action if <b>ANY</b> responding member approves the action within <b>Max. Duration Minutes</b> .                                       |
| <b>approve_majority</b>  | Request approval from the Project members and approve or deny the action based on the opinion of the <b>MAJORITY</b> of the expected responding members within <b>Max. Duration Minutes</b> .    |
| <b>approve_unanimous</b> | Request approval from the Project members and only approve the action if <b>ALL</b> of the members <b>UNANIMOUSLY</b> approve the action within <b>Max. Duration Minutes</b> .                   |
| <b>first_reply</b>       | Request approval from the Project members and approve or deny the action based on the <b>FIRST</b> responding member within <b>Max. Duration Minutes</b> .                                       |
| <b>objection</b>         | Query the Project members for 'objections' to the action and PREVENT the action if <b>ANY</b> member 'objects' to the action within <b>Max. Duration Minutes</b> .                               |
| <b>gripe</b>             | Inform the members of the Project and <b>PREVENT</b> the controlled action to proceed. <b>Max. Duration Minutes</b> is ignored.  |
| <b>silent</b>            | Silently inform the members of the Project of the action being performed without informing the user attempting the action. <b>Max. Duration Minutes</b> is ignored.                              |
| <b>tell</b>              | Inform the members of the Project of the action being performed. The user attempting the action is informed that the Project members are also informed. <b>Max. Duration Minutes</b> is ignored. |

Select a **Max. Duration** for the **Project Request Instance**. This would be the maximum time to wait for Project Member input or advice before proceeding or not with the controlled command. The range that will be allowed to select from is determined by the **Project's Minimum Duration** and **Maximum Duration** attribute values. This field is not editable and ignored if the **Request Kind** is either **Gripe**, **Silent** or **Tell**.

Provide a required **Comment**. The comment should be form 3-100 characters. The comment can contain environmental variables which are evaluated during instance execution (**\$env\_var** for UNIX and MacOS operating systems and **%env\_var%** for Windows operating systems). This comment is displayed to Project Members during notification.



Once you have completed the **Request Kind**, **Max. Duration** and **Comment** fields you can select either the **Show Client Request Command** button or **Execute Web Request** button.

**Build Project Request**

**Request Kind:** approve\_majority ▾

**Max. Duration:** 3 minutes ▾

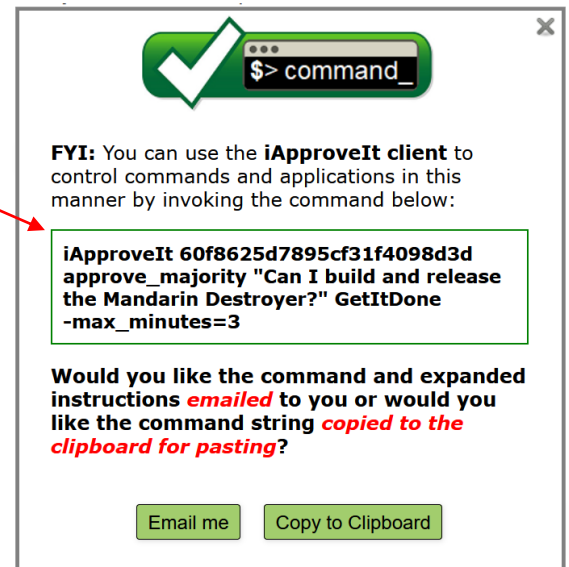
**Comment:** (required) Can I build and release the Mandarin Destroyer?

**Show Client Request Command** **Execute Web Request**

Selecting the **Show Client Request Command** button shows the **Command Line Display** dialog which displays the actual command that could be used to execute the desired policy on a client machine.

Again, this is valuable when you are first learning the commands you want to use in scripts.

Selecting the **Email Me** button will send an email with the command and instructions on its use to your account's email address. This is both helpful for ensuring that you have access to the correct command where you need it or for forwarding that email to someone that needs to implement the proper policy on your behalf.



Selecting the **Copy to Clipboard** button will copy the command to your clipboard (cut/paste buffer) so you can conveniently paste into any command window or open script you are creating.

Selecting the **Execute Web Request** button from the **Projects:Use** screen actually performs an iApproveIt web request. This is useful for testing the Project Instance, leaning about request or actually using in real-time to get advice or perform an *Associate Poll* or *Request for Advice*.

**Build Project Request**

**Request Kind:** approve\_majority ▾

**Max. Duration:** 3 minutes ▾

**Comment:** (required) Can I build and release the Mandarin Destroyer?

**Show Client Request Command** **Execute Web Request**



This is an **approve\_majority** request type sent by Tony Stark with a comment, “**Can I build and release the Mandarin Destroyer?**” sent to, the Project Members Bruce Banner, Bruce Wayne and Clark Kent.

The Project Members will get instant notifications to their *phone*, *email* or *both* and they would look similar to the one below:

Hello **Bruce**,

Please be advised, **Tony Stark** is performing an action:

***Can I build and release the Mandarin Destroyer?***

This message is of the **approve\_majority** type so, your immediate real-time permission/advise is requested for this action as one of the **3** members of the **"QA Team" Project**.

The **majority** of the Project member responses will determine if this action is allowed or not for Tony.

Comment:

(optional)

(Time Remaining to Respond = 2:40)

I Approve
I Reject

The Project Members have 3 minutes to give their approval and in this instance 2 members must **approve** the request before it is allowed.

Bruce Banner approved the request and also provided a comment of “**Yes, but Hulk smash with bare hands.**” and Bruce Wayne gave their **approval** with no additional comment. Since 2 is a majority of 3, there is no need to wait longer for Clark Kent’s response and the **Action was APPROVED**.

Associates
Projects
Configure
Logout

Create
Edit
Use

The Request was sucessfully sent.

Select to view from (2) "Self-Owned" Project(s)

(privacy) Project Name

▼

Executing - Project Request

An "approve\_majority" Request  
 With a comment of: "Can I build and  
 release the Mandarin Destroyer?"  
 Was sent to 3 members

- Bruce Banner
- Bruce Wayne
- Clark Kent

With 3 minutes to reply...

-- Any timely responses are below: --

Bruce Banner responded with an APPROVAL  
 and a comment of: Yes, but Hulk smash with  
 bare hands..

Bruce Wayne responded with an APPROVAL.

---- Connection Closed -----

| Sent | Approved | Rejected | Unresponded |
|------|----------|----------|-------------|
| 3    | 2        | 0        | 1           |

=== Action was **APPROVED** ===

Close Execution Window

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.





Now you see from this **Web Request** how a **Client Request** would work, you can use the **Client Command Request** to add similar real-time control in your programs or automated processes.

```
Command Prompt

L:\CD>iApproveIt 60f8625d7895cf31f4098d3d approve_majority "Can I build and release the Mandarin Destroyer." GetItDone -max_minutes=1

First asking these people:

    - Bruce Wayne
    - Clark Kent
    - Bruce Banner

for their real-time approval to perform this action...

    - Can I build and release the Mandarin Destroyer.

waiting up to 1 minute for the MAJORITY OF these people to decide if you can perform this action...
====
- Clark Kent APPROVES and provided a comment of:
    Yes, but Hulk smash with bare hands..
- Bruce Banner APPROVES
====
2 of the 3 members APPROVED the action.
====
Resuming with this action.
====

L:\CD>
```



## Managing Your Configuration

The purpose of the **Configure Tab** is to manage your profile, subscription and get access to help.

The **Configure:Show** screen allows you to see your Profile. The **Configure:Edit** screen allows you to modify your Profile. The **Configure:Docs** screen is where you can access iApproveIt Client Download screen and Product Documentation. Finally, the **Configure:Upgrade** screen is where you manage your iApproveIt Subscription.

### Show Your Profile

The **Configure:Show** screen displays your User Profile Specific information in read-only format. Fields that can be edited in the **Configure:Edit** screen appear as insensitive fields

|             |       |
|-------------|-------|
| First Name: | tony  |
| Last Name:  | stark |

while fields that cannot be changed appears as displayed text.

|           |                               |
|-----------|-------------------------------|
| Email:    | tony.stark@abs-consulting.com |
| Logon ID: | ironman                       |

To display the **Configure:Edit** screen to *edit* User Profile Data select the **Configure:Edit** sub-menu or select the **Edit Configuration Data** button.





## Edit Your Profile

The **Configure>Edit** screen displays your User Profile Specific information in an editable format if the field is an editable field. Fields that can be edited in the **Configure>Edit** screen appear as open-for-edit fields.

|                    |                                    |
|--------------------|------------------------------------|
| <b>First Name:</b> | <input type="text" value="tony"/>  |
| <b>Last Name:</b>  | <input type="text" value="stark"/> |

while fields that cannot be changed appears as displayed text.



|   |
|---|
| <b>Email:</b> tony.stark@abs-consulting.com |
| <b>Logon ID:</b> ironman                    |

On the **Configure>Edit** screen you can:

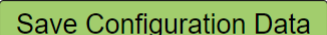
- Update your First and Last Name
- Change your Password
- Verify your mobile number.

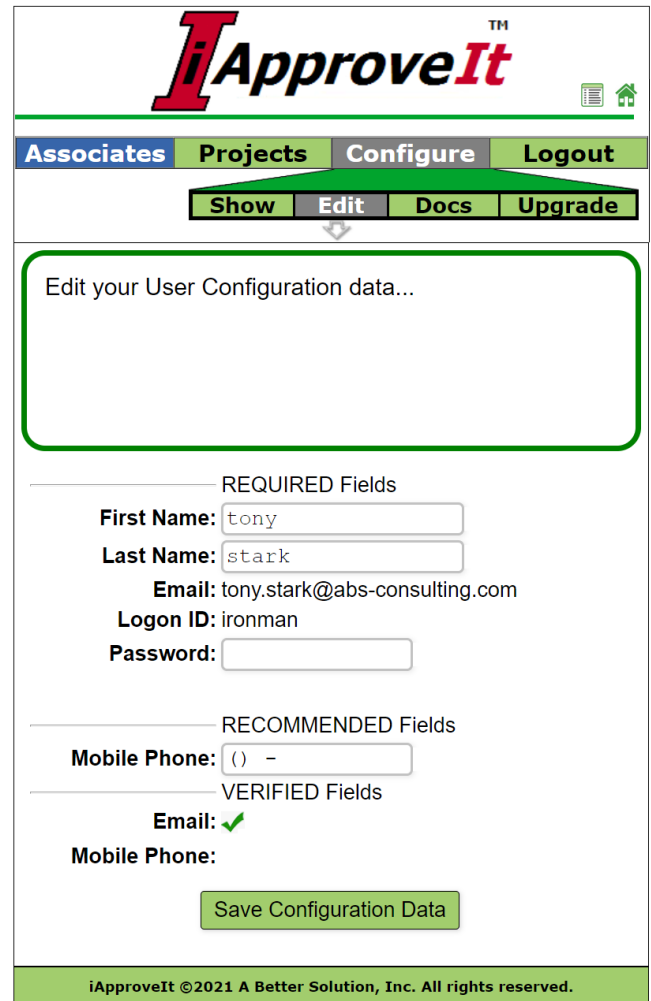
Skip and leave the password field blank if you do not wish to change it. Populate the field if you wish to change your Password.



If you populate the Password field another Password field will appear for you to type the Password in again. The Password Fields must match in order to proceed with a password Field change. Password fields are never displayed in plain text.

Passwords consist of 4-14 characters within the [a..z], [A..Z], [0..9], [+!\$?#@%^&\*\_] character sets with at least one from each character set. The second Password must match the value in the first Password field. If the passwords meet Password Criteria and match then a  will be displayed if not then an  is displayed.

Editing your Mobile Phone Number is a 2-stop process requiring verification. Refer to the next section entitled [Edit Your Profile : Verity Your Mobile Number](#) for editing.

When you have edited all fields to your liking, select the  button.



|                  |  |   |
|------------------|--|---|
| <b>Password:</b> | <input type="password" value="....."/> |  |
| <b>Password:</b> | <input type="password" value="....."/> |  |






## Edit Your Profile: Verify Your Mobile Number


Only **verified** Mobile Numbers can be saved to your Profile. This is to ensure two things:

- First, that the number is a valid phone number that can receive SMS messages
- Secondly, that the number is your number.


When you modify and leave the **Mobile Phone** field. It is first checked to confirm it a possible valid phone number by format to ensure it is a 10-digit number.


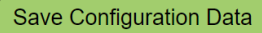
If the format is valid it is marked with a  next to it when the field focus is released if the number was previously **verified** and can be saved as part of the profile record. If the field does not contain a 10-digit number, is it marked with a  next to it. If the field is valid (a 10-digit number) AND has not yet been verified, then the “needs verification: indicator  is displayed.

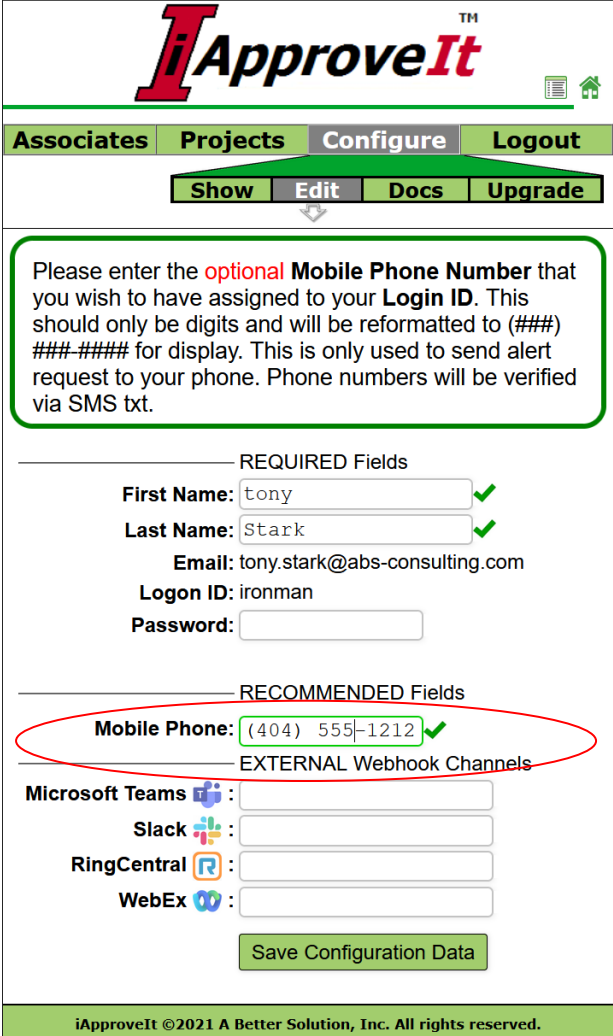
A **Mobile Phone Number** must first be **verified** before it can be saved. Select the **verify** button next to the number

Mobile Phone: (404) 555-1213 

which sends a **verification code** to that mobile phone which should be received in less than a second.

Place the code received in the **Mobile Phone Verification** dialog input field then select  button to verify the phone.

This removes the  indicator on the **Configure:Edit** screen allowing you to save the Profile data by selecting the  button.



**iApproveIt**™

Associates Projects **Configure** Logout

Show Edit Docs Upgrade

Please enter the **optional Mobile Phone Number** that you wish to have assigned to your **Login ID**. This should only be digits and will be reformatted to (###) ###-#### for display. This is only used to send alert request to your phone. Phone numbers will be verified via SMS txt.

REQUIRED Fields

First Name: tony ✓

Last Name: Stark ✓

Email: tony.stark@abs-consulting.com

Logon ID: ironman

Password:

RECOMMENDED Fields

Mobile Phone: (404) 555-1212 ✓

EXTERNAL Webhook Channels

Microsoft Teams:

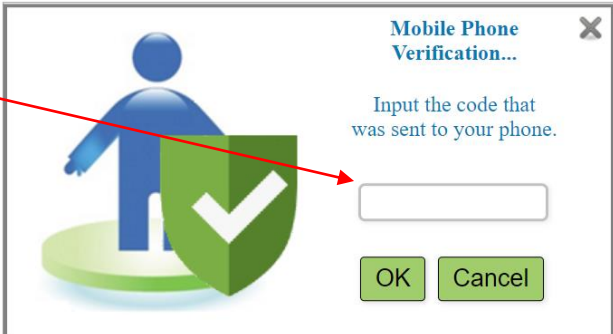
Slack:

RingCentral:

WebEx:

Save Configuration Data

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.



**Mobile Phone Verification...**

Input the code that was sent to your phone.

OK Cancel






## Edit Your Profile: Verify Your Webhook Channels



Only verified **Webhook Channel Links** can be saved to your Profile. This is to ensure two things:

- First, that the webhook address is valid and can receive incoming messages.
- Secondly, that you can indeed see the messages on the Channel.


When you modify and leave a **Webhook Channel** field. It is first checked to confirm it a possible valid secure URL link beginning with the “https:” string.


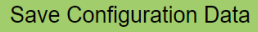
If the format is valid it is marked with a  next to it when the field focus is released if the address was previously *verified* and can be saved as part of the profile record. If the field does not contain a valid secure link string it is marked with a  next to it. If the field contains a valid secure URL link AND has not yet been verified, then the “needs verification” indicator  is displayed.

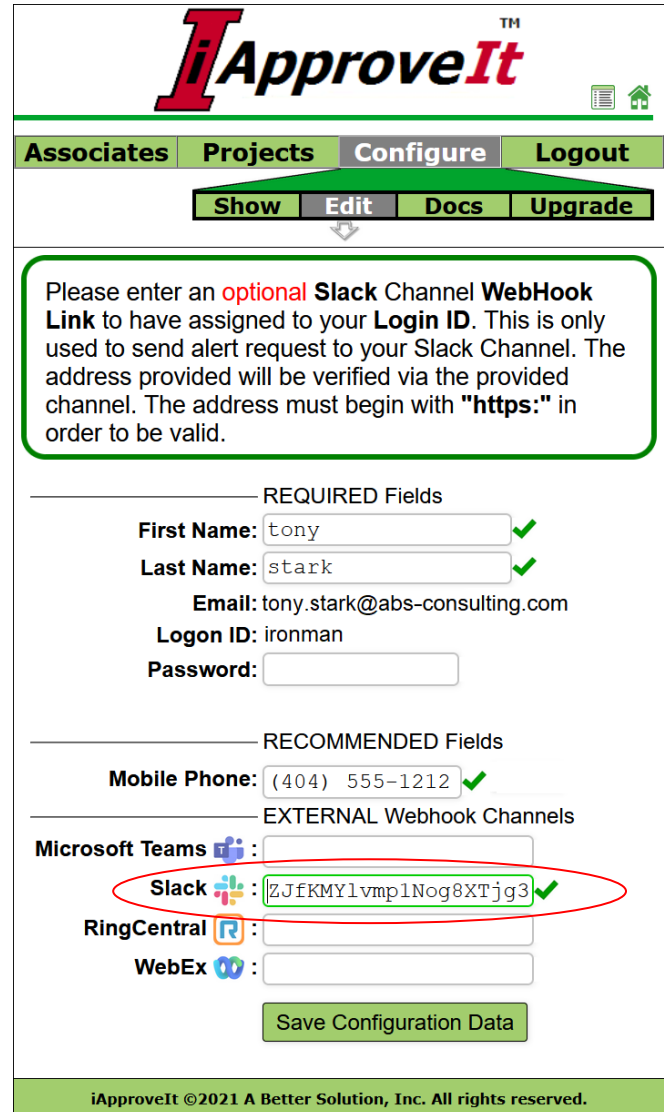
A **Webhook Channel Link** must first be verified before it can be saved. Select the **verify button** next to the link.

Slack  : https://hooks.slack. 

which sends a **verification code** via an incoming webhook to the associated service channel (i.e. Microsoft Teams, Slack, etc.) that is displayed on that channel’s interface. If the link was a proper channel link it should be displayed there in less than a second.

Place the code received in the **Webhook Link Verification** dialog input field then select  button to verify the phone.

This removes the  indicator on the Configure>Edit screen allowing you to save the Profile data by selecting the  button.




**iApproveIt™**


Associates Projects **Configure** Logout

Show Edit Docs Upgrade

Please enter an **optional Slack Channel WebHook Link** to have assigned to your **Login ID**. This is only used to send alert request to your Slack Channel. The address provided will be verified via the provided channel. The address must begin with “https:” in order to be valid.

**REQUIRED Fields**

First Name: tony 


Last Name: stark 

Email: tony.stark@abs-consulting.com


Logon ID: ironman



Password:


**RECOMMENDED Fields**


Mobile Phone: (404) 555-1212 

**EXTERNAL Webhook Channels**

Microsoft Teams  :

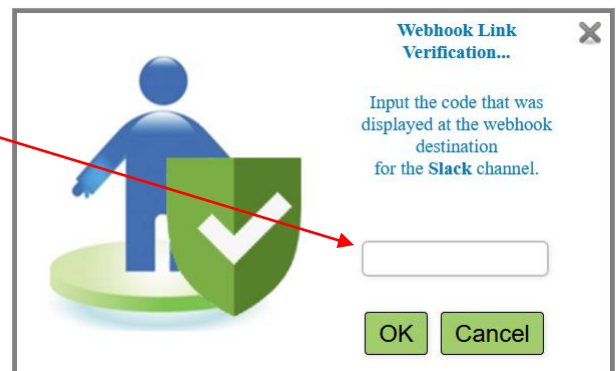
Slack  : zJfKMY1vmp1Nog8XTjg3 

RingCentral  :

WebEx  :

Save Configuration Data

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.



**Webhook Link Verification...**




Input the code that was displayed at the webhook destination for the Slack channel.

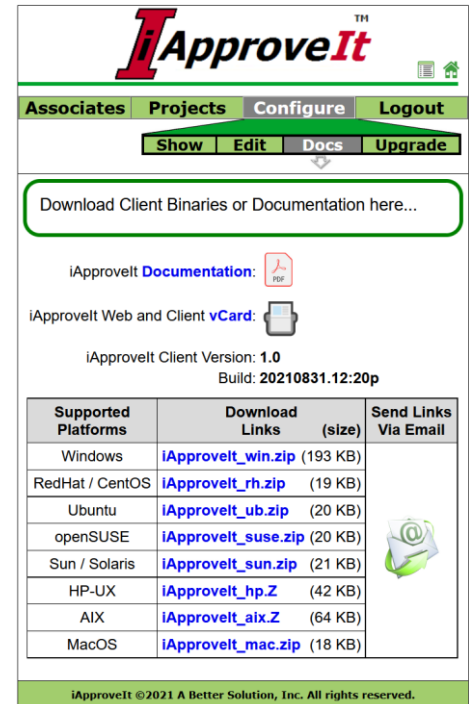
OK Cancel



## Documentation and Download

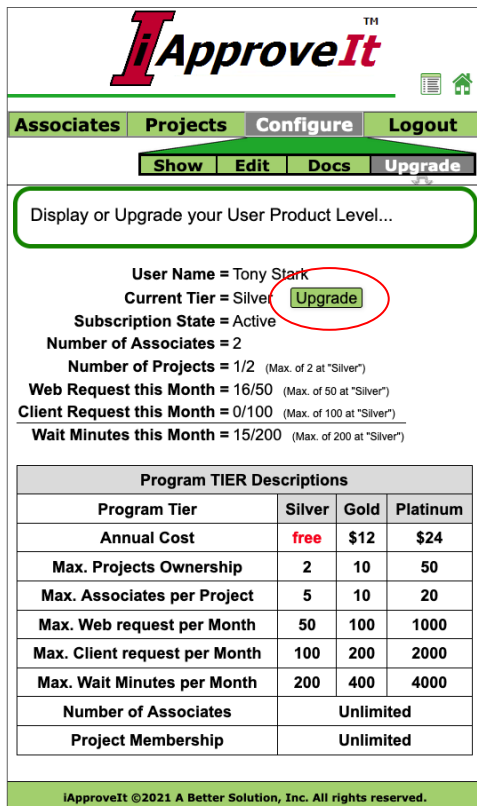
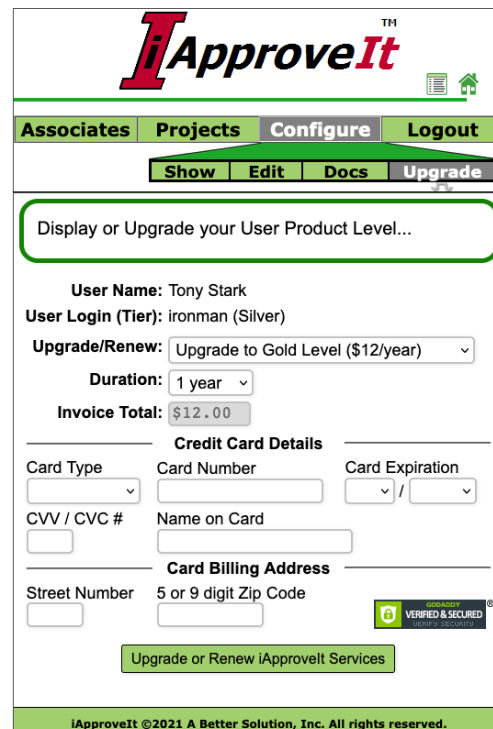
The **Docs Tab** provides documentation and iApproveIt Client downloads links for the supported client platforms. There are also additional icon links that perform other actions as summarized in the table below:

| Active Icons  | Purpose   |
|---|---|
|  | Selecting the <b>Email Links icon</b> will email the links to you at the email address you have registered.   |
|  | Selecting the <b>vCard icon</b> will you're your associated vCard registered native application (i.e. Outlook) and give you the opportunity to save the iApproveIt contact information so that iApproveIt communication can be easily identified and sorted in that application |
|  | Selecting the Documentation icon will open the iApproveIt documentation (this documentation).   |




## Upgrade Your Subscription

Under the **Configure Tab**, the Upgrade section allows you to manage your subscription. Subscriptions are per year. During the upgrade process you can choose which tier and the duration of your subscription. If your subscription is expired you will be given the option to renew or upgrade to a higher tier. At anytime you can upgrade your current subscription to the next tier.



## iApproveIt Dashboard

The iApproveIt Dashboard allows users to monitor, update and view current and previous request responses. Display the Dashboard by clicking the icon  in the upper right area of any screens.

The Dashboard is divided into three (3) sections.

- [Pending Unanswered Responses](#)
- [Pending Answered Responses](#)
- [Recently Expired & Settled Responses](#)



Each section is explained in dedicated sub-sections that follow this section.

If you are logged in to the iApproveIt Web Application, this Dashboard will automatically appear if there is anything new that requires your attention. If you close the application main window without logging off the Dashboard stays active to allow you to become aware of items that need your attention without having to run the application.

**Open Requested Responses for "Bruce Banner" (hulk)**

Pending Unanswered Responses

There are no active request pending your input or review.

Pending Answered Responses

There are no active request where you have already provided your input or review.

Recently Expired/Settled Responses

There are **11** recently expired or settled request to review within the last **8** hours.  
Select a row below to **review** response details.

| Project Name | Requesting User | Request Type | Request Comment      | Was Allowed | Your Reply | Your Advice |
|--------------|-----------------|--------------|----------------------|-------------|------------|-------------|
| Smash        | Tony Stark      | advice       | What should I do ... | yes         | yes        | Keep It     |
| Smash        | Tony Stark      | advice       | What should I do ... | yes         | yes        | Keep it     |
| Smash        | [ccclarke3]     | first_reply  | 333a                 | no          |            |             |
| Smash        | [ccclarke3]     | first_reply  | 333a                 | yes         | APPROVED   |             |
| Smash        | [ccclarke3]     | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [ccclarke3]     | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [ccclarke3]     | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [ccclarke3]     | first_reply  | ddd                  | no          |            |             |
| Smash        | [ccclarke3]     | first_reply  | ddd                  | no          |            |             |
| Smash        | [ccclarke3]     | first_reply  | ddd                  | yes         | APPROVED   |             |

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.





## Pending Unanswered Responses

When an active unanswered response is received it will populate the first section/table.

You get a summary of the request:

- Time Remaining
- Project Name
- Requesting User
- Request Type
- Request Comment

The request displayed in this section are ordered by time urgency so that those request with the least remaining time to respond are listed first.

Clicking on the request shows request details and gives you the ability to respond, if required, to the request.

The request details dialog is the same as would be displayed if you had responded from an iApproveIt SMS notification or attached vendor webhook channel.

### Open Requested Responses for "Bruce Banner" (hulk)

#### Pending Unanswered Responses

There is **1** active request pending your input or review.  
Select a row below to **provide** your input.

| Remaining Time | Project Name | Requesting User | Request Type | Request Comment      |
|----------------|--------------|-----------------|--------------|----------------------|
| 44 sec.        | QA Team      | Tony Stark      | advice       | What should I do ... |

#### Pending Answered Responses

There are no active request where you have already provided your input or review.

#### Recently Expired/Settled Responses

There are **10** recently expired or settled request to review within the last **8** hours.  
Select a row below to **review** response details.

| Project Name | Requesting User | Request Type | Request Comment      | Was Allowed | Your Reply | Your Advice |
|--------------|-----------------|--------------|----------------------|-------------|------------|-------------|
| Smash        | Tony Stark      | advice       | What should I do ... | yes         | yes        | Keep it     |
| Smash        | [cclarke3]      | first_reply  | 333a                 | no          |            |             |
| Smash        | [cclarke3]      | first_reply  | 333a                 | yes         | APPROVED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.

### Open Requested Responses for "Bruce Banner" (hulk)

#### Pending Unanswered Responses

There is **1** active request pending your input or review.  
Select a row below to **provide** your input.

| Remaining Time | Project Name | Requesting User | Request Type | Request Comment      |
|----------------|--------------|-----------------|--------------|----------------------|
| 32 sec.        | QA Team      | Tony Stark      | advice       | What should I do ... |

#### Pending Answered Responses

There are no active request where you have already provided your input or review.

#### Recently Expired/Settled Responses

There are **10** recently expired or settled request to review within the last **8** hours.  
Select a row below to **review** response details.

| Project Name | Requesting User | Request Type | Request Comment      | Was Allowed | Your Reply | Your Advice |
|--------------|-----------------|--------------|----------------------|-------------|------------|-------------|
| Smash        | Tony Stark      | advice       | What should I do ... | yes         | yes        | Keep it     |
| Smash        | [cclarke3]      | first_reply  | 333a                 | no          |            |             |
| Smash        | [cclarke3]      | first_reply  | 333a                 | yes         | APPROVED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |

**Comment:** (optional)

(Time Remaining to Respond = 31 seconds)

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.



## Pending Answered Responses

Once the request is responded to it **may** move to the second section, Pending Answered Responses, if the **request** is still in dispute.

For example, if the request was to be decided by the first member responding member (i.e. a **first\_reply** type request), then your answer would **settle** the request since only one responder need respond to settle such a request and the request would be moved to the [Recently Expired/Settled Responses](#) section.

However, if the request type was of **approve\_majority** and input from other Project Members is still required to get a majority decision and the request would be moved to the second section, the **Pending Answered Responses** section.

**Open Requested Responses for "Bruce Banner" (hulk)**

---

**Pending Unanswered Responses**

There are no active request pending your input or review.

---

**Pending Answered Responses**

There is **1** active request where you have already provided input or review. Select a row below to **change** your input.

| Remaining Time | Project Name | Requesting User | Request Type | Request Comment      | Your Reply | Your Advice |
|----------------|--------------|-----------------|--------------|----------------------|------------|-------------|
| 16 sec.        | QA Team      | Tony Stark      | advice       | What should I do ... | yes        | Keep It     |

---

**Recently Expired/Settled Responses**

There are **10** recently expired or settled request to review within the last **8** hours. Select a row below to **review** response details.

| Project Name | Requesting User | Request Type | Request Comment      | Was Allowed | Your Reply | Your Advice |
|--------------|-----------------|--------------|----------------------|-------------|------------|-------------|
| Smash        | Tony Stark      | advice       | What should I do ... | yes         | yes        | Keep it     |
| Smash        | [cclarke3]      | first_reply  | 333a                 | no          |            |             |
| Smash        | [cclarke3]      | first_reply  | 333a                 | yes         | APPROVED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |
| Smash        | [cclarke3]      | first_reply  | ddd                  | no          | REJECTED   |             |

iApproveIt ©2021 A Better Solution, Inc. All rights reserved.

Just as with the [Pending Unanswered Responses](#) section/table, you get a summary of the request:

- Time Remaining
- Project Name
- Requesting User
- Request Type
- Request Comment

| Remaining Time | Project Name | Requesting User | Request Type | Request Comment      | Your Reply | Your Advice |
|----------------|--------------|-----------------|--------------|----------------------|------------|-------------|
| 16 sec.        | QA Team      | Tony Stark      | advice       | What should I do ... | yes        | Keep It     |

As well as the **additional fields**

- Your Reply
- Your Advice

Clicking on the request shows request details (including your previous answer and comment) and gives you the ability to change your response and change or add a comment to the request.

The request details dialog is the same as would be displayed if you had responded from an iApproveIt SMS notification or attached vendor webhook channel.

Hello **Bruce**,

Please be advised, **Tony Stark** is performing an action:

**What should I do with the Tecerac?**

This message is of the **advice** type so, you are being notified of their attempt of this action as one of the **4** members of the **"QA Team" Project**.

Comment:   
(optional)

(Time Remaining to Respond = 3 seconds)

Yes

No



## Recently Expired/Settled Responses

After the response is settled or expired it moves to the third section. This shows a recent list of completed responses for your review. By clicking on any response would pull up the details of that response. Since, it is settled you cannot make changes to the response.

You see all settled responses within the last 8 hours where your response or advice was requested, whether you responded or not.

### Recently Expired/Settled Responses

There are **7** recently expired or settled request to review within the last **8** hours.  
Select a row below to **review** response details.

| Project Name | Requesting User | Request Type | Request Comment      | Was Allowed | Your Reply | Your Advice          |
|--------------|-----------------|--------------|----------------------|-------------|------------|----------------------|
| QA Team      | [cclarke3]      | objection    | Can I 'borrow' th... | no          | OBJECTED   | Absolutely Not       |
| QA Team      | [cclarke3]      | advice       | What should I do ... | yes         |            |                      |
| QA Team      | Tony Stark      | advice       | What should I do ... | yes         |            |                      |
| QA Team      | Tony Stark      | advice       | What should I do ... | yes         |            |                      |
| QA Team      | Tony Stark      | advice       | What should I do ... | yes         | yes        | Keep it safe?        |
| QA Team      | [cclarke3]      | advice       | What should I do ... | yes         | yes        | Take it for same ... |
| QA Team      | [cclarke3]      | advice       | What should I do ... | yes         | yes        |                      |

You get a summary of the request:

- Project Name
- Requesting User
- Request Type
- Request Comment
- Was Allowed
- Your Reply
- Your Advice

Hello **Tony**,

Please be advised, (**cclarke3**) is performing an action:

***Can I 'borrow' the Tecerac for the weekend?***

This message is of the **objection** type so, your immediate real-time permission/advice is requested for this action as one of the **4** members of the **"QA Team" Project**.

The action will be allowed unless any Project member **objects** to (cclarke3) performing this action.



The available **Response Time** has **expired**.

The decision was made with your timely input **considered**.

**1 of the 4 Members Objected.**





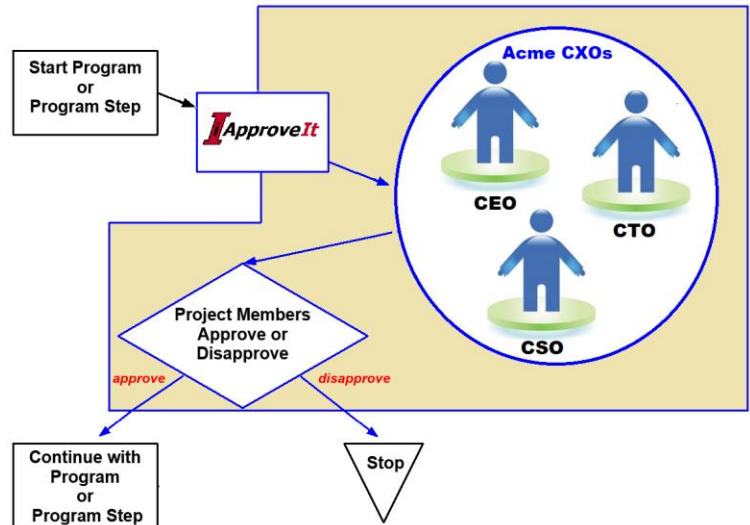
## iApproveIt (command line)

The **iApproveIt Client** program establishes communication between the process that is being controlled or audited and the devices of Project Members.

So, for any script (in any language) one will call iApproveIt when the script is at a decision or notification point. Using pseudo-code to represent any scripting language or program language, a simple script change might be *from*:

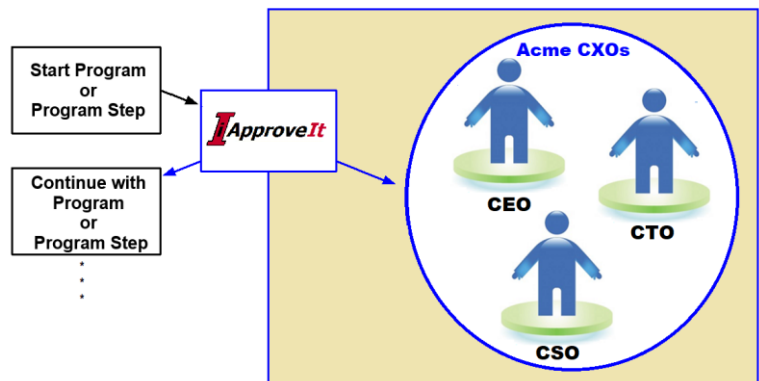
```
...
call program_to_audit
...
to:
...
return_code = call iApproveIt

if (return_code == 0)
    call program_to_audit
endif
...
```



Or for situations where just *notification* is required, one might change a script *from*:

```
...
call program_to_audit
...
to:
...
call iApproveIt
call program_to_audit
...
```





**USAGE:** iApproveIt -ver | { **project\_id** **dialog\_type** **dialog\_text** **passphrase** [-max\_minutes=minutes]  
[-proxy=host] [-port=port] [-outfile=file] [-msg\_only] [-no\_spin] }

Such that:

|                         |   |
|-------------------------|---|
| <b>-ver</b>             | produces iApproveIt <b>product version message</b>  |
| <b>project_id</b>       | is the unique <b>project_id</b> associated with the Project you wish to notify or query   |
| <b>dialog_type</b>      | represents the type of notification or query you wish to perform of the Project.<br>The valid <b>dialog_types</b> are: <ul style="list-style-type: none"> <li>• <b>advice</b></li> <li>• <b>approve_any</b></li> <li>• <b>first_reply</b></li> <li>• <b>approve_majority</b></li> <li>• <b>approve_unanimous</b></li> <li>• <b>objection</b></li> <li>• <b>gripe</b></li> <li>• <b>tell</b></li> <li>• <b>silent</b></li> </ul> <p>The <b>dialog_types</b> are outlined in section entitled: <a href="#">iApproveIt dialog types</a>.</p> |
| <b>dialog_text</b>      | Represents the text that is displayed to Project Members.   |
| <b>passphrase</b>       | Represents the passphrase required in order to successfully notify or query a project. This is required to prevent organizations from accidentally notifying or querying Projects belonging to <i>other</i> organizations.  |
| <b>-max_minutes</b>     | (optional) represents the max number of minutes a request will last before ending. This limit is bound by the Project's <b>Max. Duration</b> and <b>Min. Duration</b> settings.   |
| <b>-proxy={host}</b>    | (optional) the name of the <b>proxy host</b> if internet connection is established through a proxy. The host can also be set using the environmental variable <b>I_APPROVE_HOST</b> . The command line <b>-proxy</b> takes precedence.  |
| <b>-port={port_num}</b> | - (optional) the <b>port number</b> used for http connections. If omitted the default of port 80 is used. The port can also be set using the environmental variable <b>I_APPROVE_PORT</b> . The command line <b>-port</b> takes precedence, then the <i>environmental variable</i> (if set) then the <i>default</i> .   |
| <b>-outfile={file}</b>  | (optional) the <b>output file</b> where program <i>stdout</i> is redirected.  |
| <b>-msg_only</b>        | (optional) just displays the output of what request would have been created and sent. No actual communication with Project members is sent.   |
| <b>-no_spin</b>         | (optional) used for situations where you don't want count down information sent to STDOUT.  |

The iApproveIt client program returns zero (0) for approval or success, otherwise it returns a non-zero value. Most of the time one would only use the simplest of **iApproveIt client** calls which includes the parameters: **project\_id**, **dialog\_type**, **dialog\_text** and **passphrase**. To read more on **iApproveIt Client** usage examples refer to the section entitled: [Examples: Using the iApproveIt Client](#).



## iApproveIt dialog\_types

The **iApproveIt Client** program uses the **dialog\_type** to decide how it will communicate with the Project Members. The project administrator can create active notifications that are sent to the Project members. Those users can provide instant decision to "approve" the execution of the action or can "deny" the execution, so this action is not performed.

There are **nine (9)** iApproveIt **dialog\_types** and they fall into **three (3) categories**:

| dialog_type categories      | description  |
|-----------------------------|--|
| "Permission" or "voting"    | Where the process waits for an "approval" or "disapproval" from the voting Project Members.          |
| "Advice" request            | Where the process request and waits for "advice" from a Project Member before the process continues. |
| "Notification Only" request | Where the process informs Project members and allows the process to continue.                        |

The **nine (9)** iApproveIt **dialog\_types** fall into **three (3) categories** as depicted in the table below:

| dialog_type categories                          | dialog_type       | "Voting" Condition Required for Approval   |
|---|-------------------|--|
| "Permission" or "voting"                        | approve_any       | approval from "any" one of the project members   |
|   | approve_majority  | approval from a "majority" of the project members  |
|   | approve_unanimous | approval from "all" of the project members   |
|   | first_reply       | approval or not based on reply of first responding project member  |
|   | objection         | approved if "none" of the project members object   |
| "Advice" request                                | advice            | informs "all" of the project members that this action has completed or is about to start and queries each Project Member for "advice". It returns the "advice" of the Project Members before proceeding. It does not inhibit the invoking action.                      |
| "Advice" request<br>"Notification Only" request | gripe             | informs " <b>all</b> " of the project members that this action has was attempted and was inhibited. It <b>always inhibits</b> the invoking action.<br><br><b>Note:</b> it informs the user that the notification to Project Members is being performed.                |
|   | silent            | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>does not</b> inhibit the invoking action.<br><br><b>Note:</b> it <b>does not</b> inform the user that the notification to Project Members is being performed. |
|   | tell              | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>does not</b> inhibit the invoking action.<br><br><b>Note:</b> it informs the user that the notification to Project Members is being performed.                |



The five (5) valid "**Permission or Voting**" **dialog\_types** and their descriptions are depicted in the table below:

| <b>dialog_type</b>       | <b>"Voting" Condition Required for Approval</b>                   |
|--------------------------|---|
| <b>approve_any</b>       | approval from "any" one of the project members                    |
| <b>approve_majority</b>  | approval from a "majority" of the project members                 |
| <b>approve_unanimous</b> | approval from "all" of the project members                        |
| <b>first_reply</b>       | approval or not based on reply of first responding project member |
| <b>objection</b>         | approved if "none" of the project members object                  |

Each project can determine a minimum and maximum duration, 1 to 15 minutes, for a “decision” by the Project Members before “non-approval” is assumed. The exception being the “objection” type where, if no “objections” are received by the Project Members within the project’s duration allows the process to proceed.

The one (1) valid "**Advice**" **dialog\_types** and its description is depicted in the table below:

| <b>dialog_type</b> | <b>Condition Required for Continuation</b>  |
|--------------------|---|
| <b>advice</b>      | informs "all" of the project members that this action has completed or is about to start and queries each Project Member for "advice". It returns the "advice" of the Project Members before proceeding. It does not inhibit the invoking action. |

A duration between 1 to 15 minutes, based on project settings, is waited for “advice” by a Project Member before the process is allowed to proceed.

The three (3) valid "**Notification Only**" **dialog\_types** and their descriptions are depicted in the table below:

| <b>dialog_type</b> | <b>Notification Performed</b>  |
|--------------------|--|
| <b>gripe</b>       | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>always inhibits</b> the invoking action.<br><br><b>Note:</b> it informs the user that the notification to Project Members is being performed.                 |
| <b>silent</b>      | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>does not</b> inhibit the invoking action.<br><br><b>Note:</b> it <b>does not</b> inform the user that the notification to Project Members is being performed. |
| <b>tell</b>        | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>does not</b> inhibit the invoking action.<br><br><b>Note:</b> it informs the user that the notification to Project Members is being performed.                |



## Examples: Using the iApproveIt Client

The **iApproveIt Client** program uses the **dialog\_type** to decide how it will communicate with the Project Members. The project administrator can create active notifications that are sent to the Project members. Those users can provide instant decision to "approve" the execution of the action or can "deny" the execution, so this action is not performed.

The **nine (9)** iApproveIt **dialog\_types** fall into **three (3) categories** as depicted in the table below:

| dialog_type categories                          | dialog_type       | "Voting" Condition Required for Approval   |
|---|-------------------|--|
| "Permission" or "voting"                        | approve_any       | approval from "any" one of the project members   |
|   | approve_majority  | approval from a "majority" of the project members  |
|   | approve_unanimous | approval from "all" of the project members   |
|   | first_reply       | approval or not based on reply of first responding project member  |
|   | objection         | approved if "none" of the project members object   |
| "Advice" request                                | advice            | informs "all" of the project members that this action has completed or is about to start and queries each Project Member for "advice". It returns the "advice" of the Project Members before proceeding. It does not inhibit the invoking action.                      |
| "Advice" request<br>"Notification Only" request | gripe             | informs " <b>all</b> " of the project members that this action has was attempted and was inhibited. It <b>always inhibits</b> the invoking action.<br><br><b>Note:</b> it informs the user that the notification to Project Members is being performed.                |
|   | silent            | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>does not</b> inhibit the invoking action.<br><br><b>Note:</b> it <b>does not</b> inform the user that the notification to Project Members is being performed. |
|   | tell              | informs " <b>all</b> " of the project members that this action has completed or is about to start. It <b>does not</b> inhibit the invoking action.<br><br><b>Note:</b> it informs the user that the notification to Project Members is being performed.                |

The following sub-sections show examples of each type of **dialog\_type** within the listed categories.

- [Permission or Voting](#) - examples
- [Advice](#) - examples
- [Notification Only](#) – examples



### **“Permission” or “Voting” examples**

The **“Permission or Voting” iApproveIt dialog\_type** category holds the **dialog\_types** that allow for the “voting” of one or more Project Members before an action allowed to continue or abort. Each type has a condition that must be met before the audited action should proceed or be aborted (or just change the program state flow). The process will wait a duration between 1 to 15 minutes based on the project settings, to communicate with the Projects Members and get “responses” from those Members before a decision is made to proceed or not.

The project administrator can create active notifications that are sent to the Project members. Those users can provide instant permission to "approve" the execution of the action or can "deny" the execution, so this action is not performed.

The five (5) valid **"Permission or Voting" dialog\_types** and their descriptions are depicted in the table below:

| <b>dialog_type<br/>“Permission” or<br/>“Voting”</b> | <b>"Voting" Condition Required for Approval</b>              |
|---|--|
| <b>approve_any</b>                                  | approval from "any" one of the project members               |
| <b>approve_majority</b>                             | approval from a "majority" of the project members            |
| <b>approve_unanimous</b>                            | approval from "all" of the project members                   |
| <b>first_response</b>                               | approval from the "first" responder from the project members |
| <b>objection</b>                                    | approved if "none" of the project members object             |

The following sub-sections show examples of each type of dialog\_type within the “Permission or Voting” category.

- [approve\\_any](#) - examples
- [approve\\_majority](#) – examples
- [approve\\_unanimous](#) – examples
- [first\\_reply](#) – examples
- [objections](#) – examples



## approve\_any - examples

You wish create policy so that it requires *any* “approval” from one of the “Release Engineers” before you can run the QA\_build.bat script. You could modify your night\_build.bat script to resemble the one below:

```
----- begin: snip QA_build.bat -----
@echo off

set message="%USERNAME% wishes to run the QA build script, Is this OK?"

c:\path\to\iApproveit.exe {project_id} approve_any %message% qwerty

if %ERRORLEVEL% == 0 GOTO APPROVED

REM # None of the Project Members approved the action - abort the build
echo "QA Build aborted..."
...
GOTO END

:APPROVED

REM # One of the Project Members approved the action - continue with build
echo "QA Build started..."
...
:END
----- end: snip QA_build.bat -----
```

Now when someone (e.g., “Tony”) wants to start the “build”, which requires the approval from “any” of the “Release Engineers” for the department. They start the build script, which ask all “Release Engineers” for their approval.

If any one of the “Release Engineers” *approves*, the build *continues*...

```
c:\build_scripts>QA_build.bat
First asking these people:

- Bruce Wayne
- Clark Kent

for their real-time approval to perform this action...

- Tony wishes to run the QA build script. Is this OK?

waiting up to 1 minute for ANY ONE of these people to APPROVE
this action...
=====
- Bruce Wayne APPROVES and provided a comment of:
  Go ahead with the build.
=====
Resuming with this action.
=====
QA Build started...
```

Hello **Bruce**,

Please be advised, (**tony**) is performing an action:

**"Tony wishes to run the QA build script. Is this OK?"**

This message is of the **approve\_any** type so, your immediate real-time permission/advise is requested for this action as one of the **2** members of the **"QA Team"** Project.

**Any** Project member's approval will approve this action for .

Comment:   
(optional)

(Time Remaining to Respond = 7 seconds)





If none of the “Release Engineers” *approved* within the **request duration**, then the build is *aborted*...

```
c:\build_scripts>QA_build.bat

First asking these people:

    - Bruce Wayne
    - Clark Kent

for their real-time approval to perform this action...

    - Tony wishes to run the QA build script. Is this OK?

waiting up to 1 minute for ANY ONE of these people to APPROVE this action...
=====
- Bruce Wayne DISAPPROVES and provided a comment of:
  I'm not sure check with Tom
- Clark Kent DISAPPROVES and provided a comment of:
  QA is not ready b/c of core dump
=====
None of the 2 members APPROVED the action.
=====
Prohibiting this action.
=====
QA Build aborted...
```

Below we have implemented the *same* policy in **Perl**:

----- begin: snip QA\_build.pl -----

```
$message="$ENV{USERNAME} wishes to run the QA build script, Is this OK?";

$src = System ("iApproveIt.exe {project_id} approve_any \"$message\" qwerty");

if ($src == 0)
{
    #####
    # One of the Project Members approved the action - continue with build #
    #####

    print ("QA Build started...\n");
    ...
}
else
{
    #####
    # None of the Project Members approved the action - abort the build #
    #####

    print ("QA Build aborted...\n");
    ...
}
```

----- end: snip QA\_build.pl -----





## approve\_majority - examples

You wish create policy so that it requires a “**majority**” decision by the Members of the “**Company CXOs**” Project before you can run the **Disperse\_Checks.pl** script. “Majority” being defined as:

$$\text{floor} (\frac{1}{2} * \text{number of Project members send message}) + 1$$

so that this table applies:

| For these number of Members<br>in the Project | A “Majority” is<br>defined as |
|---|-------------------------------|
| 1   | 1                             |
| 2   | 2                             |
| 3   | 2                             |
| 4   | 3                             |
| 5   | 3                             |
| 6   | 4                             |
| 7   | 4                             |
| 8   | 5                             |
| 9   | 5                             |
| and so on...                                  | and so on...                  |

You could modify your Disperse\_Checks.pl script to resemble the one below:

----- begin: snip Disperse\_Checks.pl -----

```
$message="$ENV{USERNAME} wishes to disperse Payroll checks, Is this OK?";

$src = System ("iApproveIt.exe {project_id} approve_majority \"$message\" qwerty");

if ($src == 0)
{ #####
  # A "majority" of the C-Level executives approved - cut the checks #
  #####

  print ("Cutting Payroll Checks...\n");
  ...
}
else
{ #####
  # No "majority approval" of the CXOs attained - hold the checks #
  #####

  print ("Sorry, holding Payroll Checks...\n");
  ...
}
```

----- end: snip Disperse\_Checks.pl -----

Now when someone wants to "**Disperse Checks**", which requires a “**majority**” approval from the “**Company CXOs**". They start the Disperse\_Checks.pl script, which ask all "Company CXOs" for their approval.



The process will wait until a “**majority**” reached or the maximum wait time is reached before returning to either *continue* (cut the checks) or *abort* (hold the checks).

```
==>Disperse_Checks.pl

First asking these people:

- Bruce Wayne
- Clark Kent
- Tony Stark

for their real-time approval to perform this action...

- Jarvis wishes to disperse Payroll checks, Is this OK?

waiting up to 1 minute for the MAJORITY OF these people to decide if you can perform this action...
=====
- Bruce Wayne APPROVES
- Tony Stark APPROVES and provided a comment of:
  Make mine the biggest
=====
2 of the 3 members APPROVED the action.
=====
Resuming with this action.
=====
Cutting Payroll Checks...
```

If a “**majority**” approval could not be attained in the *required duration* or a Majority of Members “**reject**” the action as to not allow for a “**majority**” approval, then the iApproveIt Client returns a non-zero which *prevents* the checks from being cut.

```
==>Disperse_Checks.pl

First asking these people:

- Bruce Wayne
- Clark Kent
- Tony Stark

for their real-time approval to perform this action...

- Jarvis wishes to disperse Payroll checks, Is this OK?

waiting up to 1 minute for the MAJORITY OF these people to decide if you can perform this action...
=====
- Bruce Wayne DISAPPROVES
- Tony Stark APPROVES and provided a comment of:
  Make mine the biggest.
- Bruce Wayne DISAPPROVES
=====
Only 1 of the 3 members APPROVED the action.
=====
Prohibiting this action.
=====
Holding Payroll Checks...
```



## approve\_unanimous - examples

You wish create policy so that it requires a “*unanimous*” decision by the Members of the “**Company CXOs**” Project before you can run the **Disperse\_Checks.pl** script.

You could modify your Disperse\_Checks.pl script to resemble the one below:

----- begin: snip Disperse\_Checks.pl -----

```
$message="$ENV{USERNAME} wishes to disperse Payroll checks, Is this OK?";

$src = System ("iApproveIt.exe {project_id} approve_unanimous \"$message\" qwerty");

if ($src == 0)
{ #####
  # A "unanimous" of the C-Level executives approved - cut the checks #
  #####

  print ("Cutting Payroll Checks...\n");
  ...
}
else
{ #####
  # No "unanimous approval" of the CXOs attained - hold the checks #
  #####

  print ("Sorry, holding Payroll Checks...\n");
  ...
}
```

----- end: snip Disperse\_Checks.pl -----

Now when someone wants to "Disperse Checks", which requires a “*unanimous*” approval from the “**Company CXOs**”. They start the Disperse\_Checks.pl script, which ask all "Company CXOs" for their approval.

The process will wait until a “*unanimous approval*” is reached or the maximum wait time is reached before returning to either *continue* (cut the checks) or *abort* (hold the checks).

```
==>Disperse_Checks.pl

First asking these people:

- Bruce Wayne
- Clark Kent
- Tony Stark

for their real-time approval to perform this action...

- Jarvis wishes to disperse Payroll checks, Is this OK?

waiting up to 1 minute for the UNANIMOUS agreement of ALL of these people to decide if you can
perform this action...
====
- Tony Stark APPROVES
- Bruce Wayne APPROVES
- Clark Kent APPROVES
====
3 of the 3 members APPROVED the action.
====
Resuming with this action.
====
Cutting Payroll Checks...
```



If a “*Unanimous Approval*” could not be attained in the **required duration**, then the **iApproveIt Client** returns a non-zero which prevents the checks from being cut.

```

==>Disperse_Checks.pl

First asking these people:

    - Bruce Wayne
    - Clark Kent
    - Tony Stark

for their real-time approval to perform this action...

    - Jarvis wishes to disperse Payroll checks, Is this OK?

waiting up to 1 minute for the UNANIMOUS agreement of ALL of these people to decide if you can perform this action...
====
    - Tony Stark DISAPPROVES
    == End-of 1 minute maximum wait... ==
====
None of the 3 members APPROVED the action.
====
Prohibiting this action.
====
Holding Payroll Checks...

```

### first\_reply - examples

You wish create policy so that it requires a decision by a **Member** of the “**Release Engineers**” Project before you can run the **QA\_build.sh** script, you decide that the “*first*” responder will be the decision maker. You could modify your night\_build.bat script to resemble the one below:

```

----- begin: snip QA_build.sh -----
#!/bin/ksh

message="`whoami` wishes to run the QA build script, Is this OK?"

iApproveIt {project_id} first_reply "$message" qwerty`

if [ "$?" -ne 0 ]
then
    #####
    # One of the Project Members approved the action - continue with build #
    #####

    echo "QA Build started..."
    ...
else
    #####
    # None of the Project Members approved the action - abort the build #
    #####

    echo "QA Build aborted..."
    ...
fi
----- end: snip QA_build.sh -----

```



Now when someone (e.g., “Tony”) wants to start the "build", which requires the approval from the “first” responding “**Release Engineer**” for the department. They start the build script, which ask all "Release Engineers" for their approval.

If the first responder *approves*, the build *continues*...

```
==>QA_build.sh

First asking these people:

- Bruce Wayne
- Clark Kent

for their real-time approval to perform this action...

- Tony wishes to run the QA build script. Is this OK?

waiting up to 1 minute for the FIRST RESPONDING person to decide
if you can perform this action...
====
- Bruce Wayne APPROVES
====
Resuming with this action.
====
QA Build started...
```

Hello **Bruce**,

Please be advised, (**tony**) is performing an action:

**Tony wishes to run the QA build script. Is this OK?**

This message is of the **first\_reply** type so, your immediate real-time permission/advise is requested for this action as one of the **2** members of the **"QA Team"** Project.

The **first** Project member that respond will determine if this action ia allowed or not for .

Comment:

(optional)

(Time Remaining to Respond = 49 seconds)

If none of the “Release Engineers” *approved* within the **required duration** or the “*first*” responder “*denied*” the request, then the build is *aborted*...

```
c:\build_scripts>QA_build.vbs

First asking these people:

- Bruce Wayne
- Clark Kent

for their real-time approval to perform this action...

- Tony wishes to run the QA build script.. Is this OK?

waiting up to 1 minute for the FIRST RESPONDING person to decide if you can perform this action...
====
- Clark Kent DISAPPROVES
====
None of the 2 members APPROVED the action.
====
Prohibiting this action.
====
QA Build aborted...
```



## objection - examples

You wish create policy so that it requires a “*no objections*” decision by the Members of the “QA Managers” Project before you can run the **Release\_Product.sh** script

You could modify your Release\_Product.pl script to resemble the one below:

----- begin: snip Release\_Product.pl -----

```
$message="$ENV{USERNAME} wishes to release the iApproveIt Software, Is this OK?";

$src = System ("iApproveIt.exe {project_id} objection \"$message\" qwerty");

if ($src == 0)
{ #####
  # There were no objections registered In the 60 seconds - release the hounds #
  #####

  print ("Releasing the iApproveIt Software...\n");
  ...
}
else
{ #####
  # There was an objection registered - hold the release. #
  #####

  print ("Hold the release of the iApproveIt Software...\n");
  ...
}
```

----- end: snip Release\_Product.pl -----

Now when someone wants to "Release Software", which is allowed so long NONE of the “QA Managers” OBJECT, they start the Release\_Product.pl script, which ask all "QA Managers" if they have objections.

The process will wait until there is one “*objection*” by a “QA Manager” or until the maximum wait time is reached before returning to either *continue* (there were no objections) or *abort* (there was an objection).

```
==>Release_Product.pl

First asking these people:

- Bruce Wayne
- Clark Kent
- Tony Stark

if they have any real-time objection to you performing this action...

- Jarvis wishes to release the iApproveIt Software, Is this OK?

If NO ONE objects within 1 minute you will be able to proceed with this action...
====
- Tony Stark APPROVES
- Bruce Wayne OBJECTED
====
One of the 3 members OBJECTED.
====
Prohibiting this action.
====
Hold the release of the iApproveIt Software...
```



If there were “*No Objections*” during the *required duration* or *all* Members “approve” the action then the **iApproveIt Client** returns a zero which allows the “Release” to *proceed*.

```
==>Release_Product.pl

First asking these people:

  - Bruce Wayne
  - Clark Kent
  - Tony Stark

if they have any real-time objection to you performing this action...

  - Jarvis wishes to release the iApproveIt Software, Is this OK?

If NO ONE objects within 1 minute you will be able to proceed with this action...
====
  - Tony Stark APPROVES
  - Bruce Wayne APPROVES
  - Clark Kent APPROVES
====
None of the 3 members OBJECTED to you performing the action.
====
Resuming with this action.
====

Releasing the iApproveIt Software...
```





### “Advice” examples

The “Advice” iApproveIt **dialog\_type** category holds the single **dialog\_type** “advice” that allows for the passing of advice from Project Members to end users. The “advice” **dialog\_type** does not stop any action it will just wait a duration between 1 to 15 minutes, based on project settings, for advice from the **Project Members** before proceeding or not.

The one (1) valid "Advice" **dialog\_types** and its description is depicted in the table below:

| dialog_type<br>"Advice"<br>request | Condition Required for Continuation   |
|------------------------------------|---|
| <b>advice</b>                      | informs "all" of the project members that this action has completed or is about to start and queries each Project Member for "advice". It returns the "advice" of the Project Members before proceeding. It does not inhibit the invoking action. |

A duration between 1 to 15 minutes, based on project settings, is waited for “advice” by a Project Member before the process is allowed to proceed.

### advice - examples

You wish create policy so that prior to a certain action or program step you ask for advice from Members of a Project. In this case you want to make sure you know who the client’s salesman is before proceeding with the client order. So, before the proceeding in a certain step in the order process the user ask the Members of the “Sales Team” Project “Who is the salesman for this account”.

Project before you execute a certain step in the **Process\_Order .pl** script you poll the “Sales Team”. You could modify your Process\_Order.pl script to resemble the one below:

----- begin: snip Process\_Order.pl -----

```
$message="$ENV{USERNAME} is about to process an order, who is the salesmen for
$Client_name?";
```

```
$src = System ("iApproveIt.exe {project_id} advice \"$message\" qwerty");
```

```
if ($rc == 0)
{ #####
# take advantage of the advice provided and proceed with the order #
#####

print ("Using advice... if provided ..\n");
...
}
```

----- end: snip Process\_Order.pl -----



Now when someone wants to "Process in Order", which requires that all "Sales Team" members are notified and asked which one is responsible for this sale. They can make use of the Real-time advice which is printed to the command windows terminal. Optionally, this "advice" can be stored in a file or variable and use to taste, but in general iApproveIt returns a return code equal to zero and does itself not ultimately affect whether or not the calling program or program part continues.

```
==>Process_Order.pl

First asking these people:

  - Bruce Wayne
  - Clark Kent
  - Tony Stark

for their real-time advice on performing this action...

  - Jarvis is about to process an order, who is the salesmen for Evil Corp?

waiting up to 1 minute for their advice...
====
- Clark Kent APPROVES and provided a comment of:
  It's me.
== End-of 1 minute maximum wait... ==
====
Any advice and suggestions provided. You can continue.
====
Resuming with this action.
=====
```

If "**No Advice**" was provided during the maximum wait time then the **iApproveIt Client** still returns a return code equal to zero and does itself not ultimately affect whether or not the calling program or program part continues.

```
==>Process_Order.pl

First asking these people:

  - Bruce Wayne
  - Clark Kent
  - Tony Stark

for their real-time advice on performing this action...

  - Jarvis is about to process an order, who is the salesmen for Evil Corp?

waiting up to 1 minute for their advice...
====
== End-of 1 minute maximum wait... ==
====
Any advice and suggestions provided. You can continue.
====
Resuming with this action.
=====
```



### “Notification Only” examples

The “*Notification Only*” iApproveIt **dialog\_type** category holds the **dialog\_types** that allow for “one-way notifications” to be sent to one or more Project Members before an action allowed to continue or has been completed. The “notifications” are sent to the Project Members before a decision is made to proceed or no with the audited process, program or program step.

The three (3) valid “**Notification Only**” **dialog\_types** and their descriptions are depicted in the table below:

| dialog_type<br>"Advice" request<br>"Notification Only"<br>request | Notification Performed  |
|---|---|
| <b>gripe</b>  | <p>informs "<i>all</i>" of the project members that this action has completed or is about to start. It <i>always inhibits</i> the invoking action.</p> <p><b>Note:</b> it informs the user that the notification to Project Members is being performed.</p>                 |
| <b>silent</b>   | <p>informs "<i>all</i>" of the project members that this action has completed or is about to start. It <i>does not</i> inhibit the invoking action.</p> <p><b>Note:</b> it <i>does not</i> inform the user that the notification to Project Members is being performed.</p> |
| <b>tell</b>   | <p>informs "<i>all</i>" of the project members that this action has completed or is about to start. It <i>does not</i> inhibit the invoking action.</p> <p><b>Note:</b> it informs the user that the notification to Project Members is being performed.</p>                |

The following sub-sections show examples of each type of **dialog\_type** within the “Notification Only” category.

- [gripe](#) – example
- [tell](#) - examples
- [silent](#) - example



## gripe - example

You wish create policy certain users are prevented from performing a certain action or program step and you want the “**ABS Admins**” to be notified that the attempt was made. Specifically, you want to prevent anyone accept “**cclarke**” or “**rcarter**” from performing an **iBark** services shutdown on a server. You also want the “**ABS Admins**” to be notified is some unauthorized user attempts to do perform the service shutdown

You could modify your **Shutdown\_iBark\_Service.pl** script to resemble the one below:

----- begin: snip Shutdown\_iBark\_Service.pl -----

```
$operator = $ENV{USERNAME};

if ((" $operator" ne "cclarke3") && (" $operator" ne "rcarter"))
{
    $message="the user \" $operator\" was prevented from stopping the iBark service.";
    system ("iApproveIt.exe {project_id} gripe \" $message\" qwerty");

    exit (1);
}

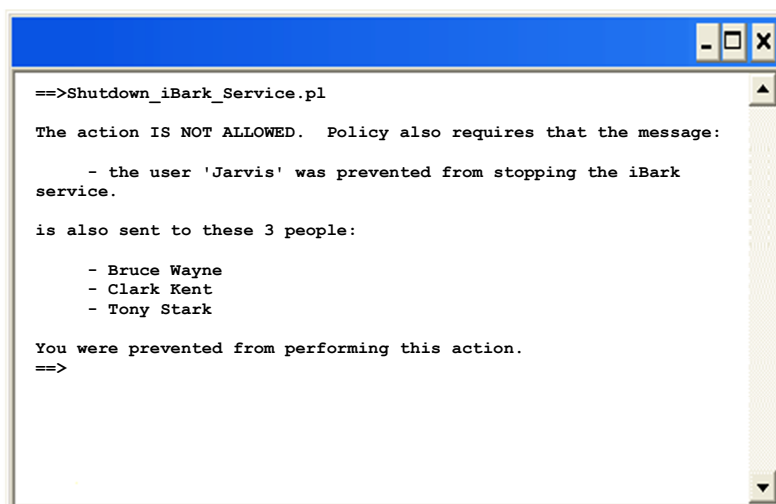
else
{
    print ("Shutting down iBark services on the server \"n");

    ...

    print ("Completed the shutdown of iBark services on the server.\"n");
}
```

----- end: snip Shutdown\_iBark\_Service.pl -----

Now when an unauthorized user attempts to "stop the iBark service", all “**ABS Admins**” are notified and the attempting user is preventing from stopping the service and also informed that the ABS admins were notified.



Hello **Clark**,

Please be advised, (**jarvis**) attempted to perform the action:

*the user 'Jarvis' was prevented from stopping the iBark service.*

This is a **gripe** type message, so they were prevented from performing the action and you are being notified of their attempt of this action as one of the **3** members of the "**QA Team**" Project.



## tell – examples

You wish create policy so that prior to a certain action or program steps you inform the “**Company CXOs**” that this action or program step is about to be performed. You specifically want the “Company CXOs” to be notified anytime there is a “*press release*” so they are always in-the-know.

You could modify your **Press\_Release.pl** script to resemble the one below:

----- begin: snip Press\_Release.pl -----

```
$message="ACME has just made this announcement \"$announcement_title\".";
system ("iApproveIt.exe {project_id} tell \"$message\" qwerty");
print ("Posting Press Release \"$announcement_title\" to Twitter.\n");
```

----- end: snip Press\_Release.pl -----

Now when someone wants to "publish a prepared Press Release", all C-Level Executives are notified.

```
==>Press_Release.pl

This action IS ALLOWED. Policy also requires that the message:

- ACME has just made this announcement 'ABS Introduces iApproveIt 1.0'.

is also sent to these 3 people:

- Bruce Wayne
- Clark Kent
- Tony Stark

You can proceed with this action.

Posting Press Release "ACME has just made this announcement 'ABS Introduces iApproveIt 1.0'" to Twitter.
```

Each member of the project would receive a notification like this one.

Hello **Clark**,

Please be advised, **Tony Stark** is performing an action:

**ACME just made this announcement 'ABS introduces iApproveIt 1.0'**

This message is of the **tell** type so, you are being notified of their attempt of this action as one of the **3** members of the **"QA Team" Project**.

You could just as easily modify your **Press\_Release.pl** script so send the messages *after* a successful press release as depicted below:

----- begin: snip Press\_Release.pl -----

```
print ("Posted the Press Release \"$announcement_title\" to Twitter.\n");
$message="ACME has just made this announcement \"$announcement_title\".";
system ("iApproveIt.exe {project_id} tell \"$message\" qwerty");
```

----- end: snip Press\_Release.pl -----



## silent - example

You wish create policy so that prior to a certain action or program steps you inform the “**ABS Admins**” that this action or program step is about to be performed. You specifically want the “ABS Admins” to be notified anytime the “Website Update” script is run. ***You do not however want the user to know that the “ABS Admins” are being notified.***

You could modify your **Website\_Update.pl** script to resemble the one below:

----- begin: snip Website\_Update.pl -----

```
print ("Starting the update of the \"production\" ABS website...\n");

$message="$ENV{USERNAME} has started the update for the Production Website.";

system ("iApproveIt.exe {project_id} silent \"$message\" qwerty");

print ("Completed the update of the \"production\" ABS website.\n");
```

----- end: snip Website\_Update.pl -----

Now when someone "updates the production website", all “**ABS Admins**” are notified without *the knowledge of the invoking user*.



Notice, no indication was given that the “ABS Admins” were notified, even though they all received a notification as they had with a “tell” message

Hello **Tony**,

Please be advised, (**cclarke3**) is performing an action:

**cclarke3 has started the update for the 'production' website**

This message is of the **silent** type so, you are being **silently** notified of their attempt of this action as one of the **3** members of the **"QA Team" Project**.

You could just as easily have made the “silent” call after the program step. (Or both before and after).



### **Misc. or “Multi-call” examples**

The iApproveIt Product can be used in countless ways. A process can have any number of calls to the **iApproveIt Client** to match the program or process Flow chart the organization wishes to emulate. Examples of iterative/interactive processes can be viewed in the subsection entitled:

[“Hierarchical – examples”](#).

The iApproveIt Product can be integrated with countless other programs, scripts or commercial off-the-shelf software packages. A non-exhaustive list products that iApproveIt has been integrated with can be viewed in the subsection entitled: [“Other Product Integration – examples”](#).

The following sub-sections show other examples of iApproveIt use:

- [Hierarchical policy](#) – example
- [Other Product Integration](#) – examples





## “Hierarchical Policy” - example

You wish create policy so that it requires a “**majority**” approval from the “QA\_Managers” and if they provide a “**majority approval**” then it requires an *additional* approval from the “CQO” (Chief Quality Officer). If the “QA Managers” **cannot** form a “majority approval” the CQO is not disturbed. If the “QA Managers” for a “majority approval” and the CQO still rejects it then the “QA Managers are notified that the CQO *override* their “approval” and “rejected” the release.

You could modify your Release.pl script to resemble the one below:

----- begin: snip Release.pl -----

```
$operator = $ENV{USERNAME};

$message = "the user \"$operator\" wants to release the product without completing the \
full test plan. Is this OK?";

$src = system ("iApproveIt.exe {project_id} approve_majority \"$message\" qwerty");

if ("{$src} eq "0")
{
    #####
    # There was a "majority Approval" from the QA Managers, now ask the CQO #
    #####

    $message = "The QA Managers are OK with the product being released without \
    completing the full test plan, but this still requires your approval. \
    Are you also OK with this?";

    $src = system ("iApproveIt.exe {project_id} approve_unanimous \"$message\" qwerty");

    if ("{$src} eq "0")
    {
        #####
        # Also approved by CQO - continue with risky release #
        #####

        print ("Continuing with release...\n");
        ...
        print ("Complete release...\n");
    }
    else
    {
        print ("Request to release without completing a full test plan denied by CQO.");

        $message = "Sorry I had to override the \"QA Manager\" consensus, but releasing \
        without completing this test plan is too risky.";

        system ("iApproveIt.exe {project_id} tell \"$message\" qwerty");
    }
}
else
{
    print ("Request to release without completing a full test plan denied by QA Managers.");
}
```

----- end: snip Release.pl -----



## **“Other Product Integration” - examples**

The iApproveIt product is not vendor specific, it can be used with any script or program or any program that uses or can use scripts, triggers or other definable actions for automation.

ABS has used iApproveIt with a list of products to include but not limited to:

- IBM Rational ClearCase and ClearCase Multisite (or any SCM product that can make use of scripts or return codes)
- IBM Rational ClearQuest (or any Defect tracking product that can make use of scripts or return codes)
- IBM Rational Team Concert
- IBM Rational Build Forge (or any build automation tool like Maven, Cruise Control, AntHill, etc).
- IBM Rational Functional Tester
- IBM WebSphere MQ (previously known as “MQ Series”)
- Git, GitLab, GitHub
- HP OpenView
- SAP
- MS Excel Spreadsheet automation
- MS Quickbooks
- CA Unicenter
- Vtiger, Siebel, SugarCRM, Salesforce.com (and other CRM Solutions)
- ClearTrigger, ClearReplica and iDeployIt by ABS
- ClearCheck by GoMidjets
- Windows “AT” jobs - MAC and UNIX “cron” jobs
- Other Workflow Automation Tools

ABS has used iApproveIt to provide real-time approval and/or notification using the programming languages include but not limited to:

|                         |       |             |                |
|-------------------------|-------|-------------|----------------|
| C, C++, C#, Objective-C | Perl  | AppleScript | Visual Basic   |
| ksh, sh, csh, bash, tk1 | Batch | Java        | PowerBuilder   |
| PHP                     | Make  | Cobol       | many others... |