# eSolia TrackStudio Design and Setup QuickGuide

*You are reading a QuickGuide for TrackStudio, the intent of which is to give an overview of some of the major concepts in TrackStudio as a supplement to the manuals and support forums, with the hope of assisting during the initial phases of your TrackStudio setup.*

*While evaluating TrackStudio as a task tracking system for my company which performs support and project management for clients, I tried to organize the large number of questions I asked and the answers I received into written format, for the purpose of disseminating basic information within my own organization eSolia, but also as a record for myself and possible help document for other users.*

*This document is a generalized form of those notes, and owes a debt to Maxim Kramarenko whose tireless support of TrackStudio users is exemplary, and very much appreciated.*

*—Rick Cogley*

*eSolia Inc. (http://www.esolia.com)*

*Tokyo, Japan*

## Rules and Constraints

TrackStudio is a powerful system, and so has some rules and constraints which stem from its design. After reading and digesting this section, the user documentation may begin to make more sense. It may take a few reviews of this to really understand the power of TrackStudio.

### Power of Inheritance

TS uses the power of inheritance to make it possible to make settings at a top level of the hierarchy, and have child objects automatically inherit those settings. The entire program is based on this concept and in general child objects cannot have greater permissions or rights than that of their parent.

### Special TrackStudio Vocabulary

In TS, the concept of a role or group is called a “Status.” The administrator status is built-in, and you can create others to assign permissions to. “Workflows” are the sets of states, transitions, custom fields and priorities that you combine together for performing similar tasks. “Categories” are selectable workflow instances which you make available for selection in the TS folder structure, and from which you create a folders or tasks there.

After you create an object, for example by selecting a category like “Issue” or “Bug” in a support folder, you then can append “Messages” or comments to it. Who can enter a message and how, is controlled by how you set up the workflow underlying the category.

### User Hierarchy

First, an "entity root user" must be created under the TS root user to hold each entity's main users and any of their subordinate users. Note that “child” users default to having the same or lower permissions than their parent user, and that only a parent user can manage its child users. The “entity root user” in the Sample Inc database is John Smith.

Next the users under the entity root user can be created, and there are two ways to do it. Either you can treat them as department heads, if your organization is organized by function, in which case they can then create and control their subordinate users. Or, if you have a “flatter” organization which operates as a matrix, you may find that TrackStudio will respond well to a flat setup with all users under the entity root user on the same “level” in the hierarchy. Either way, log in as the entity root user when you create them.

An artifact of TrackStudio's design is that subordinate users inherit permissions. Therefore you can use the following trick to allow some flexibility in who can manage client users: that is, you can create a "dummy" user called “Client Users” at the same level in the hierarchy as your main users. Create some subordinate users acting as placeholders for companies such as “Acme Inc Users”. Then, under each of those, create each client's named users – John Doe et al. Further, you can create subordinate users to John Doe, or, John can log in and do it himself.

Now one can log in as the “Client Users” user, create and edit client users, and even disable the entire group of client users by disabling the “Client Users” user, or, for a single company by disabling the company placeholder user.

One mechanical tip - when creating users, the login name cannot be changed from within TS without a delete and recreate (although it can be changed with a SQL query). Therefore, it is best to decide the pattern and stick with it for internal users. TS seems to *allow* spaces in usernames, but scripts and reports sometimes might respond negatively to this. Therefore, we recommend creating usernames with a consistent style such as first-initial-last-name: *jsmith, jdoe* and so on.

The concept of a root user and subordinate managers in the user hierarchy protects you from having to see and deal with all available permissions, categories and workflows in the system, in every single tab, when logged on as root. This is especially imporant if you are operating in a situation like TrackStudio's TrackStudio Host service, which has 1000 or more subordinate managers (“entity root users”), about 350 workflows and 150 statuses. If you had set all the workflows and statuses as the main root user, it would be *impracticable* to perform normal activities when logged in as root due to the sheer possible number of combinations.

The Sample Inc company's “entity root user” is John Smith (jsmith/jsmith). John Smith, for example, cannot edit the default “Folder” workflow since it was created by the TS root user and is linked to an inaccessible root task. Further, John cannot change the permissions for the administrator (all permissions granted) status. That said, John can create his own statuses and subordinate users, which allows him to create the required functionality for users and then just “set and forget” allowing the subordinate users to in turn create *their* own statuses and subordinate users.

There is a caveat. Once subordinate users are created, however, another manager cannot manage their settings or permissions. This would be the case if two managers handle the same group of subordinates, say on different shifts, or, when two sales managers have to manage the same group of client users. This constraint will be addressed in the next version of TS, which is due to be released in fall 2005.

In Summary:

- Only parent users can manage their children.
- Set up all users directly under the entity root user in a matrix organization.
- “Trick” the system by creating a dummy user to log into, from which you can control subordinate users.
- The user login id cannot be changed from within TS once users are created, but the “view name” can.
- Two managers cannot manage the same group of subordinate users in the hierarchy.

## Permissions via “Statuses”

Before permissions can be set in any detail in the system, you need to create statuses while logged in as the entity root user (not TS root). Statuses can be thought of as “permission sets” that you can assign to a user or users in one step. The display of available statuses in permissions tabs for categories and workflows is dependent upon the logged-in user. Logged in as TS root, you will see only the administrator status as available for selection. However, you can log in as the entity root user, and if you have created statuses using this user, you will see them all in the permissions tabs.

See the Sample Inc database, logged in as John Smith, for some status examples. Statuses can include functional business designations like “Department Managers” or “Engineers” or “External Clients”, but also can be related to TS activities – such as “TS: Task Admin” or “TS: Script Admin”. Further, in true TrackStudio fashion, you can set up a hierarchy of statuses – parent and child – with a child status inheriting no more than the permissions of the parent status.

Once you create a status, on its permissions tab there is a detailed list of permissions available for that status, such as “Can upload files” or “Can create users”. Furthermore, statuses are cumulative. Your “effective status” in TS is the sum of the statuses you have been granted. If a user has two statuses A and B, has a YES for permission A in status A, and a NO for the same in status B, the result is a YES.

In summary:

- Statuses are like permission sets that can be assigned to users.
- Statuses can be job functions – manager, developer – or TS activities – task admin, script admin.
- Permissions are cumulative, and your effective permissions come from the combination of your statuses.

## The Result: Matrix Management

The result of the above concepts and TS functions is that users can have multiple bosses in the system based on the user management hierarchy, and the statuses and project access control tab for a particular folder. A matrix is created in this way, where sometimes you are able to manage permissions and settings, and sometimes not based on your subordinate status in a particular hierarchy.

## Workflows and Categories

Starting with the top level Projects #1 object, you create sub-folders here to structure your task hierarchy and you do this by choosing categories which have been set up in that folder. Categories are the way you select Workflows in TS. Defining the two:

- Workflows—groupings of:
  - Priorities
  - States (open or closed)
  - Messages (what you use in TS to move from state to state, and what allows you to add a thread of comments to work)
  - Transitions (the links between states)
  - Permissions (what statuses can use what message types)
  - Customizations (custom fields for you to make the workflow hold the data you need)
- Categories—the *application* of the workflow to a folder and statuses. Which folders is the workflow used in, and by whom can it be used.

Starting with the built-in Folder workflow, you can start creating other workflows. For example:

| ***Workflow*** | ***Usage*** | ***Example Custom Fields*** |
|---|---|---|
| Folder: Contract | Holds subfolders and tasks for engagement contracts. | Start Date, Contract Number, Support Hours, Contact Person etc |
| Folder: Company | Holds company's contracts | Address, Tel, Fax, Map etc |
| Issue | Used in a contract folder for tracking issues through their lifecycle – new, open, working, resolved etc | Impact, Affected System etc |
| Risk | Used in a contract folder for tracking open risks through their lifecycle – newly identified, confirmed, mitigated etc | Affected Business Area, Current vs Future, Affected Projects etc |

Now categories can be created to allow the assignment of the workflows to a folder in the task hierarchy. Some key points to note:

- Mark which categories are available as sub-tasks in the Category Edit tab. Remember this one, since it was not so intuitive during the initial setup.
- Set which statuses can create, edit, handle and delete tasks created from the category in the Category Permissions tab.
- Set which statuses can view, process and handle which message types within a workflow in the Workflow Permissions tab.
- Notice that there is a difference between logging in as root vs logging in as a manager subordinate to root. If your statuses have been created logged in as the “entity root user” then they will only show up in the above tabs when logged in as that user.

A little work setting this up will lead you to understand that TrackStudio is a very flexible system, which can be made to work for various types of organizations that need to track tasks. It's not just for software development companies.

As expected there are some constraints and caveats:

- You can edit a task only when you are the handler, and, the default handler of the parent task becomes the default handler of the child. However, this can be configured (Categories->Permissions, "Can Edit").
- If you have created a custom workflow field, and then later delete it, TS will follow your orders and delete ALL existing instances of that field for all relevant data. Use extreme caution when changing these fields, and, be sure everything is set up properly before use.
- If a category is already being used by some tasks, then you cannot switch workflows for the category.

# Base TS Structure

There are three areas of structure to be concerned with during setup:

1. Users
2. Statuses
3. Task Folders

Since it is possible but perhaps not so easy to make changes to the structure once you are using it, you'll want to put the effort in up front. I asked a lot of questions about this, to find out what structure might be "best" for my company. The following concept shows what we have discovered and decided so far.

## Users

Assuming your organization is a matrix, you can set your users up "flat", with a dummy user for managing client users. For example, Tim Wilson might report to Rick Cogley in the organization, but in TrackStudio you set the two users up on the same level. If you are sure a user can be subordinated to, and will always be managed by another user, you can set them up like Enzo Gotti is in this illustration, subordinated to Kathy Russo.

- TrackStudio (built-in)
  - User Management (built-in)
    - TS Root User (built-in)
      - Sample Inc. Root User (built-in)
      - MyCo Inc Root User
        - !MyCo Client Users
          - ACME Users
            - |Kathy Russo
              - Enzo Gotti
          - ABC Users
            - John Gerber
          - XYZ Users
            - etc
        - Tim Wilson
        - Rick Cogley
        - etc
      - Other Entity Inc. Root User

Use this user to manage client users.

Can disable this user to disable access for all subordinates.

## Statuses

Although it is possible to structure TS statuses based on departments, a smaller company like mine can be set up with basic job titles for statuses, like so:

| Status | Owner | Parent |
|---|---|---|
| 1000 Dep't Manager | MyCo Inc Root User | administrator |
| 1100 Sr. Project Manager | MyCo Inc Root User | 1000 Dep't Manager |
| 1110 Project Manager | MyCo Inc Root User | 1100 Sr. Project Manager |
| 1200 Sr. Consultant | MyCo Inc Root User | 1000 Dep't Manager |
| 1210 Consultant | MyCo Inc Root User | 1200 Sr. Consultant |
| 1300 Sr. Engineer | MyCo Inc Root User | 1000 Dep't Manager |
| 1310 Engineer | MyCo Inc Root User | 1300 Sr. Engineer |
| 1400 Sr. Designer | MyCo Inc Root User | 1000 Dep't Manager |
| 1410 Designer | MyCo Inc Root User | 1400 Sr. Designer |
| 1500 Sr. Support | MyCo Inc Root User | 1000 Dep't Manager |
| 1510 Support | MyCo Inc Root User | 1500 Sr. Support |
| 1600 Sr. SW Developer | MyCo Inc Root User | 1000 Dep't Manager |
| 1610 SW Developer | MyCo Inc Root User | 1600 Sr. SW Developer |
| 1620 SW Tester | MyCo Inc Root User | 1600 Sr. SW Developer |
| 1900 Assistant | MyCo Inc Root User | 1000 Dep't Manager |
| 3100 Sr. Sales Rep | MyCo Inc Root User | 1000 Dep't Manager |
| 3110 Sales Rep | MyCo Inc Root User | 8100 Sr. Sales Rep |
| 3200 GA Staff | MyCo Inc Root User | 1000 Dep't Manager |
| 3210 Fin/Acct Staff | MyCo Inc Root User | 1000 Dep't Manager |
| 8000 Client | MyCo Inc Root User | 1000 Dep't Manager |
| 9001 TS: User Admin | MyCo Inc Root User | administrator |
| 9001 TS: Script Admin | MyCo Inc Root User | adminstrator |

## Task Folders

When you log in, see the folder tree on the left side of the interface. Based on experimentation and the way TS functions, we ended up setting up our structure similar to this, conceptually:

- TrackStudio (built-in)
    - Task Management (built-in)
        - Projects #1 (built-in)
            - Sample Inc. (built-in)
            - MyCo Inc.
                - INBOX
                - Engagements
                    - ACME
                        - INBOX
                        - Support
                        - Project A
                    - ABC
                        - INBOX
                        - Support
                    - XYZ
                        - INBOX
                        - Support
                    - etc
                - Internal
                    - eSolia System Admin
                    - eSolia Project X
                - Public
                    - FAQ / Knowledgebase
            - Legal Entity 2

Will use this folder to capture any mail from “unknown users” after a spam check.

Holds all folders that are going to be accessible by both MyCo and client users.

Controlled access here, and each folder has an import rule for grabbing incoming email.

The inbox for each client captures email that is not specific to any one project.

Internal only projects here.

Any user can access the KB or FAQ, and tasks here can be linked to tasks entered for support.

## Notes

- The folder tree cannot be sorted. Instead, select from Task Management link at top of main panel. Additionally, for large task lists, each user can elect to show only top level folders. See user properties.
- The top of the folder tree is built-in folders that cannot be changed or deleted. Sub-folders of Projects #1 are editable.

# TS Structure Details

Let's take a look at how the folders and workflows might fit together for MyCo's client "XYZ", zooming in on part of the folder hierarchy. Note the designation to show what category and workflow are active:

**[[ C: Category Name - W: Workflow Name ]]**

- XYZ Co. **[[ C: Folder for Companies - W: Folder: Companies ]]**
  - Inbox **[[ C: Inbox - W: Folder ]]**

    XYZ User default folder. Mail from registered users sent to the general TS address with no special subject goes here. No special import rules.

    - *Issue – X does not work* ***[[ C: Issue - W: Issue ]]***
    - *Issue – Y is broken*
    - *...*
  - Contract 123 (Master Contract) **[[ C: Folder for Contracts - W: Folder: Contracts ]]**
    - Statement of Work A (for Support) **[[ C: SoW - W: SoW ]]**
      - Support **[[ C: Support - W: Activity ]]**

        The support and emergency folders will be set to have email import rules that expect to find a different string per folder on incoming mail.

        - Emergency **[[ C: Emergency - W: Activity ]]**
          - *Issue—The Sky Is Falling!...* ***[[ C: Issue - W: Issue ]]***
        - *Issue—Can't do this or that...* ***[[ C: Issue - W: Issue ]]***
      - Regular Work **[[ C: Regular - W: Activity ]]**

        Separate folders so that different people can receive notifications or subscribe to get regular updates.

        - *Visit—20050707* ***[[ C: Visit - W: Visit ]]***
        - *System Check—20050710* ***[[ C: System Check - W: Checklist: System ]]***
        - *Outage—20050702* ***[[ C: Outage Report - W: Planned Maintenance ]]***
        - *Request—20050703* ***[[ C: Request - W: Request ]]***
      - Meetings **[[ C: Meetings - W: Folder ]]**
        - *Mtg – Re Problem X 20050702* ***[[ C: Meeting - W: Meeting ]]***
      - Projects **[[ C: Projects - W: Activity ]]**
        - *A Support-related Project* **[[ C: Project - W: Activity ]]**
      - Guidelines / Decisions **[[ C: Guideline - W: Folder ]]**
        - *Guideline – Network standard* ***[[ C: Guideline - W: Guideline ]]***
        - *Guideline – HW standard*
        - *Policy –Do's and Don't's*
    - SoW B (for Project) **[[ C: SoW - W: SoW ]]**
      - Emergency Inbox **[[ C: Emergency - W: Activity ]]**

        The emergency folder will be set to have an email import rules that expects to find a different string on incoming email.

        - *Issue—Consultants AWOL!...* ***[[ C: Issue - W: Issue ]]***
      - Project Phases, Activities and Tasks **[[ C: Projects - W: Activity ]]**
        - Phase 1 – Initiate
          - SubActivity

            The project activity folders will be set to have email import rules that expect to find a different string per folder on incoming mail.

            - *Issue—Problem with system setup...* ***[[ C: Issue - W: Issue ]]***

    - Phase N – Topic
        - *Issue—Phase N Issue..****[[ C: Issue - W: Issue ]]***
- Risks **[[ C: Risks - W: Folder ]]**
    - *Risk—Migration Problems...****[[ C: Risk - W: Risk ]]***
- Q&A **[[ C: Q&A - W: Folder ]]**
    - *Q&A—What is the policy on...****[[ C: Q&A - W: Q&A ]]***
- Meetings **[[ C: Meetings - W: Folder ]]**
    - Tactical **[[ C: Meetings - W: Folder ]]**
        - *Mtg – Process Discussion 20050702* ***[[ C: Meeting - W: Meeting ]]***
    - Review
        - *Mtg – Sponsor Meeting 20050703* ***[[ C: Meeting - W: Meeting ]]***
- Guidelines / Decisions **[[ C: Guideline - W: Folder ]]**
    - *Guideline – Conferencing Center Numbers* ***[[ C: Guideline - W: Guideline ]]***
    - *Guideline – Accounting Standards*
    - *Policy –Do's and Don't's*

## Folder Structure Options

One thing to note about structuring your TS projects and tasks hierarchy is that there are a number of different ways you can set it up. For example, you could create a folder to hold tasks with different workflows – visits, system checks or outages:

- Regular Work
  - *Visit*
  - *System Check*
  - *Outage*

.... or, you can separate out each folder:

- Regular Work
  - Visits
    - *Visit YYMMDD*
  - Checks
    - *Check YYMMDD*
  - Outages
    - *Outage YYMMDD*

You should create separate folders if:

- You have many objects (filters, workflows, categories, reports) specific to each separate folder – Visits, Checks, Outages.
- You need to define different default handlers for each folder.
- You want to allow only some users to view the contents of each folder. It is critical to note that you cannot hide tasks in version 3.1 based on their category, workflow or status – instead you should group them into folders if this is required. The ability to control "view" permissions on per-category basis will be added in a forthcoming version, due to be released in Fall 2005.

## Workflows

Set up as many workflows as are needed to accurately represent your operations, each with different priorities, states, messages, transitions and custom fields. A support organization might need:

- Folder
- Folder: Company
- Folder: Contract
- Folder: Statement of Work
- Activity
- Visit
- Meeting
- Issue
- Risk
- Checklist

... and so on, for starters.

## One Example: Issue Workflow

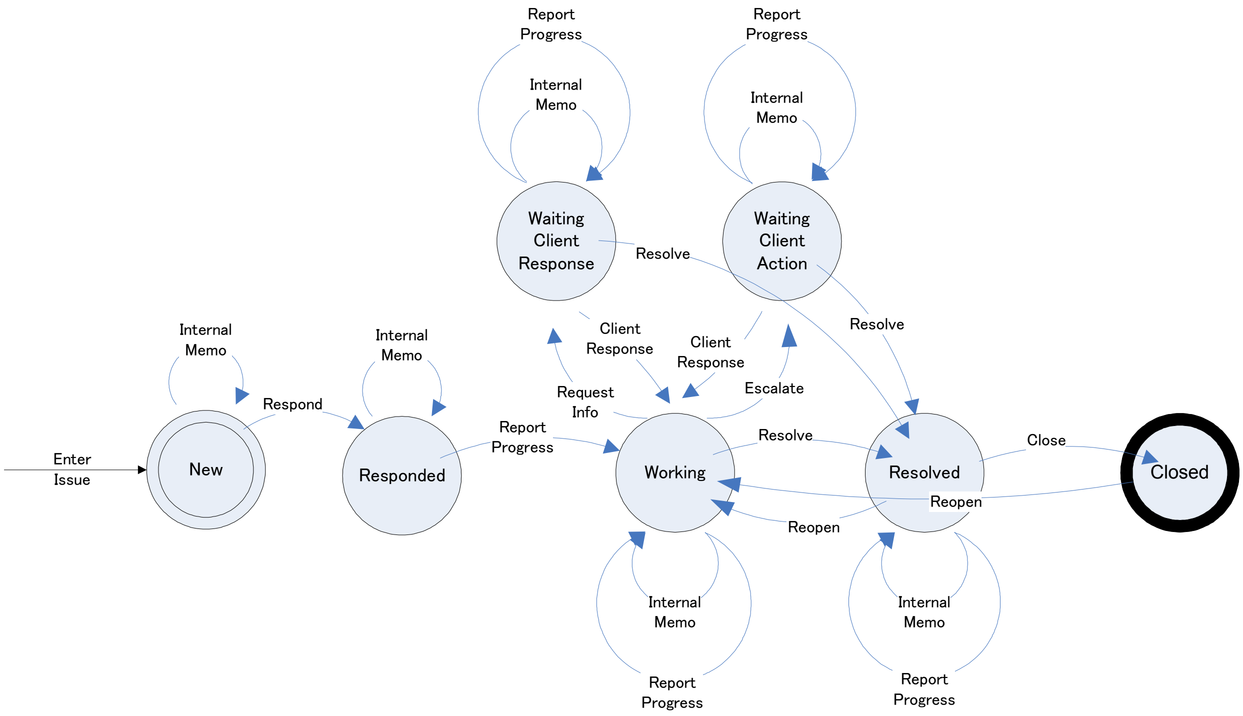
Let's take the opportunity to detail the Issue workflow a little bit more, for example.

### Priorities

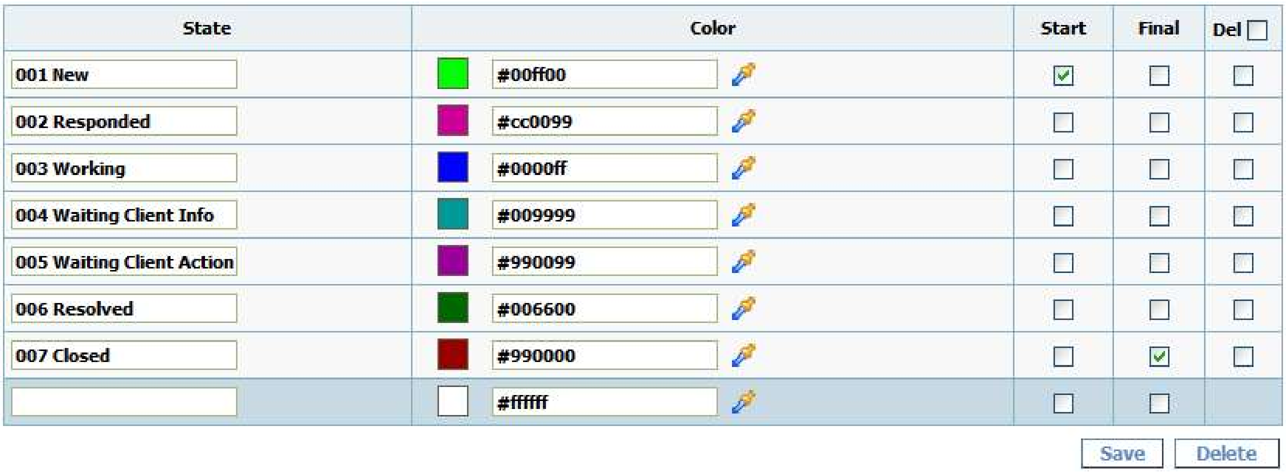
Set them to whatever is appropriate, but one way is: High, Normal, and Low, defaulting to Normal.

### States, Messages, Transitions

If you think of a workflow diagram like this one, states are the circles, and messages and transitions are the arrows between the circles. The intention here is that the "internal memo" type is only going to be visible to MyCo users, but "Report Progress" will be used to summarize for the client.

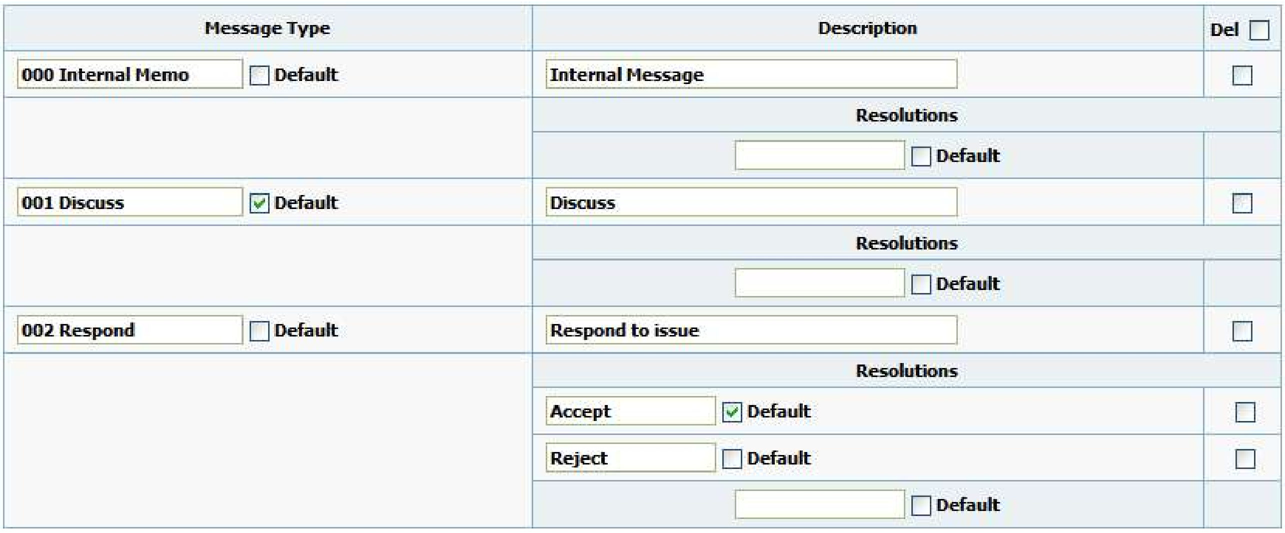


Setting this up in TrackStudio starts with States. Note that you might want to sort the States by pre-pending numbers such as in this screen shot, and assign colors to give a visual cue as to what Issue is in what State, as well as indicating Start and Final States.



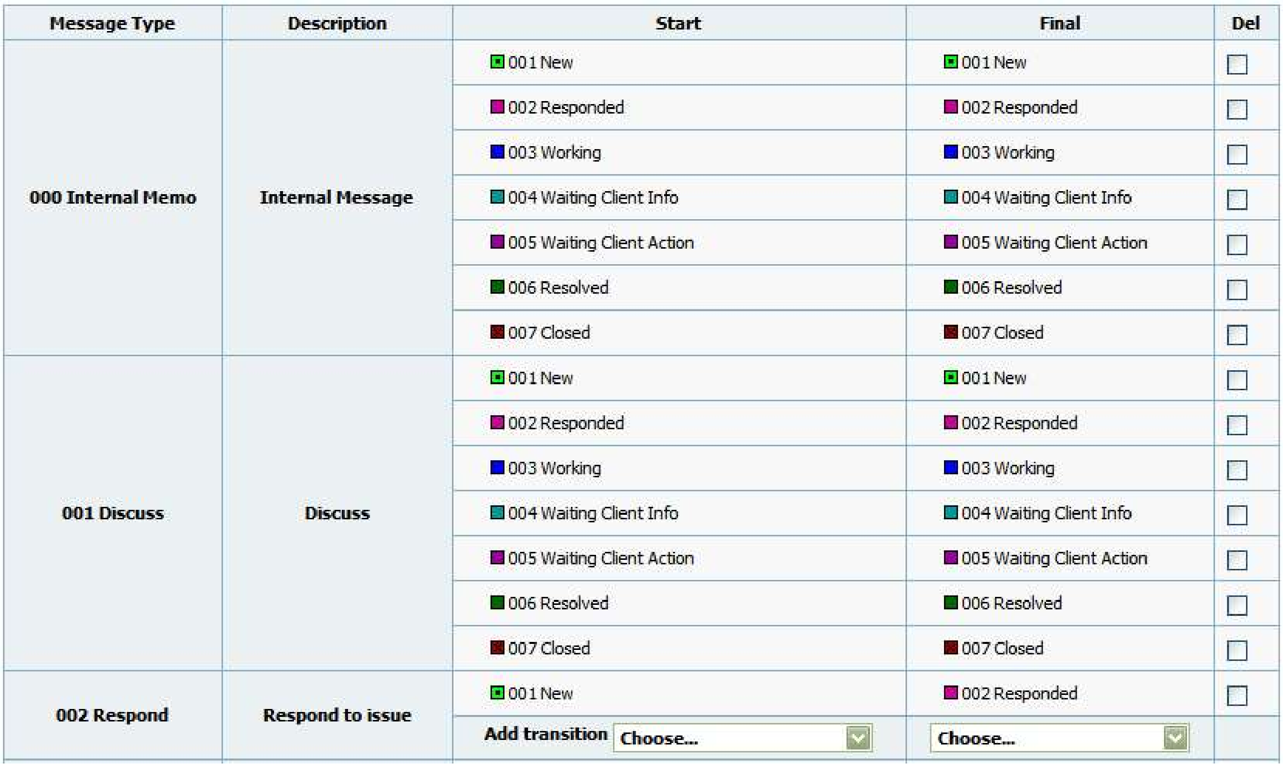
| State | Color | | Start | Final | Del □ |
|---|---|---|---|---|---|
| 001 New | 🟩 | #00ff00 | ☑ | □ | □ |
| 002 Responded | 🟪 | #cc0099 | □ | □ | □ |
| 003 Working | 🟦 | #0000ff | □ | □ | □ |
| 004 Waiting Client Info | 🟦 | #009999 | □ | □ | □ |
| 005 Waiting Client Action | 🟪 | #990099 | □ | □ | □ |
| 006 Resolved | 🟩 | #006600 | □ | □ | □ |
| 007 Closed | 🟥 | #990000 | □ | ☑ | □ |
| | ⬜ | #ffffff | □ | □ | |

Message Types can then be set up. Here is a partial screenshot, and you can see that I added another type “Discuss” which I intended for use on every state, for discussion between anyone involved. This would “loop back” in a similar way in the workflow diagram above, to the Internal Memo and Report Progress types. You can also set resolutions for each message type, which you can think of as “marker” text for each state—more of a resolution to that state, and not to the whole process.



| Message Type | Description | Del □ |
|---|---|---|
| 000 Internal Memo □ Default | Internal Message | □ |
| | Resolutions | |
| | □ Default | |
| 001 Discuss ☑ Default | Discuss | □ |
| | Resolutions | |
| | □ Default | |
| 002 Respond □ Default | Respond to issue | □ |
| | Resolutions | |
| | Accept ☑ Default | □ |
| | Reject □ Default | □ |
| | □ Default | |

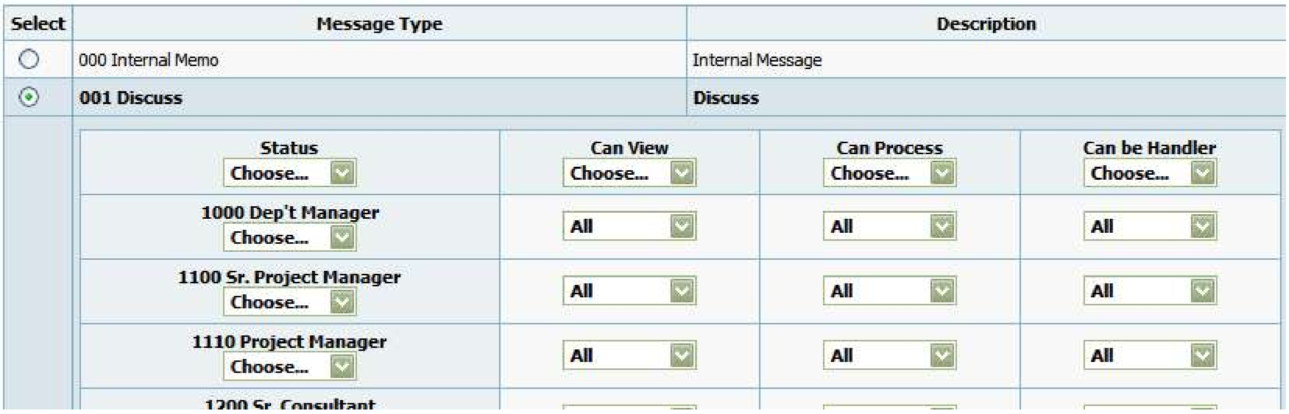
Now you can add transitions for each message type to the mix. For “loopback” messages just add the same state in Start and Final. Other message types you might want to limit—in the way we have limited “002 Respond” in this set of transitions. Respond is only for going from the New state, to the Responded state. No other types are possible between those two states.



| Message Type | Description | Start | Final | Del |
|---|---|---|---|---|
| 000 Internal Memo | Internal Message | 001 New | 001 New | □ |
| | | 002 Responded | 002 Responded | □ |
| | | 003 Working | 003 Working | □ |
| | | 004 Waiting Client Info | 004 Waiting Client Info | □ |
| | | 005 Waiting Client Action | 005 Waiting Client Action | □ |
| | | 006 Resolved | 006 Resolved | □ |
| | | 007 Closed | 007 Closed | □ |
| 001 Discuss | Discuss | 001 New | 001 New | □ |
| | | 002 Responded | 002 Responded | □ |
| | | 003 Working | 003 Working | □ |
| | | 004 Waiting Client Info | 004 Waiting Client Info | □ |
| | | 005 Waiting Client Action | 005 Waiting Client Action | □ |
| | | 006 Resolved | 006 Resolved | □ |
| | | 007 Closed | 007 Closed | □ |
| 002 Respond | Respond to issue | 001 New | 002 Responded | □ |
| | | Add transition Choose... | Choose... | |

## Permissions

Next you set permissions for each Message Type. Here you can get into considerable complexity if you have a lot of statuses, totaling number of Message Types times the number of Statuses. Check the Sample Inc database for some examples of how the permissions on the Message Types can be set, restricting the use of certain Message Types to certain statuses only.

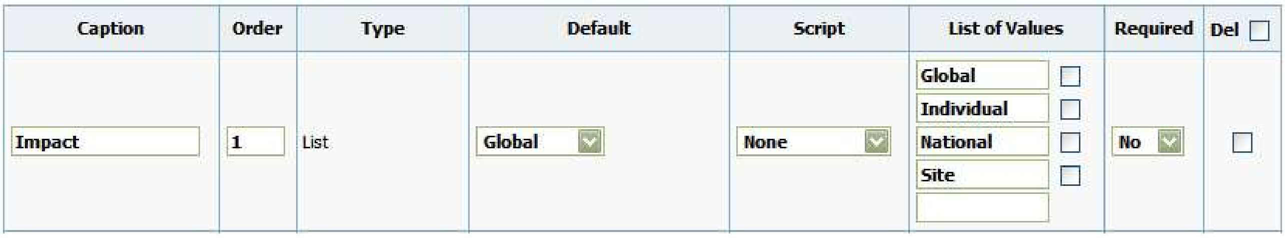


| Select | Message Type | Description |
|---|---|---|
| ○ | 000 Internal Memo | Internal Message |
| ● | **001 Discuss** | **Discuss** |

| Status<br>Choose... | Can View<br>Choose... | Can Process<br>Choose... | Can be Handler<br>Choose... |
|---|---|---|---|
| 1000 Dep't Manager<br>Choose... | All | All | All |
| 1100 Sr. Project Manager<br>Choose... | All | All | All |
| 1110 Project Manager<br>Choose... | All | All | All |
| 1200 Sr. Consultant | | | |

### Customizations

Finally, for each workflow you can create custom fields, that reference and run scripts as needed. Here we create the “Impact” list, for our users to choose what the impact of a certain issue is—Global, National, Site, or Individual.



| Caption | Order | Type | Default | Script | List of Values | Required | Del □ |
|---|---|---|---|---|---|---|---|
| Impact | 1 | List | Global | None | Global □<br>Individual □<br>National □<br>Site □ | No | □ |

## In Conclusion

Thank you for taking the time to read this document. We hope it will help you during the design and setup phase.