

The Core Protocols V. 1.0

TO: *Readers*

FR: *Jim and Michele McCarthy, McCarthy Technologies, Inc.*

RE: *The Core Protocols Version 1.0 Distribution*

Dear Recipient:

The information in this document, The Core Protocols V. 1.0, is “software for your head.” Although it is a somewhat new take on how to distribute ideas, we think the medium of software, its common characteristics, and the way people view it, is the best way to distribute any system of functional ideas and procedures. People understand that software is supposed to do something, and they also understand that successful software will likely be upgraded with better (or at least different) versions over time. You will find these things to be true of The Core protocols.

There are many other benefits to treating procedures as a kind of software and distributing them as software is distributed; but, whatever benefits there may be, they all depend on the customer receiving a perfectly faithful copy of the program to begin with. This document is your copy.

With computers, when you install a new program on your hard drive, you reasonably expect the files provided by the manufacturer to be a replica of the master program. Though the program may have other problems, most often the fidelity of the bits involved in your copy is not of great concern to you. With software for your head, things are a bit different. The document you are reading now is meant to support the person-to-person “copying technology” on which we depend for the promulgation of The Core protocols.

If someone handed you these pages prior to discussing The Core protocols with you, or e-mailed them to you, you should be aware that he or she is only behaving in accordance with his or her obligations under The Core Licensing Agreement. This License is an instance of the “GNU General

Public License” (www.fsf.org/copyleft/gpl.html), and, as such, it governs the free distribution of The Core protocols, among other things.

The Core Protocols V. 1.0, distributed in its entirety within these few pages, consists of

- 1. This letter*
- 2. The Core protocols*
- 3. The Core Protocols License Agreement*

These three elements constitute (wholly and inseparably) The Core Protocols V. 1.0. Later versions may well be available on our Web site (www.mccarthy-tech.com). The files mentioned in the License Agreement (below) consist of these three parts. All three of these parts should be included with the package you are now reading. Please note that there are many other pieces of The Core that you might find useful, especially the patterns, antipatterns, and definitions of The Core. These are published in a book¹ (wherein these words were also first published), but they are not included here, and are not governed by The Core license.

When you distribute The Core protocols, what you distribute are the pages in this Appendix. If you decide to tell someone about The Core protocols, or to “teach” them to someone, you are, in fact, copying The Core protocols. Such verbal “copying” efforts are buggy and incomplete. They can also be very annoying, especially if the person doing the copying is evangelical. But we do think it is good to copy The Core protocols, so we’ve made it free of charge to do so, provided that you actually copy it: Copying The Core protocols means that you must provide all the material in this document before or at the time you attempt to impart any of the substance of The Core protocols to someone else. You must include (1) This statement from us, (2) The Core protocols, and (3) The Core Protocols License Agreement, either in writing or by e-mail (an electronic version of this document is found on our Web site at www.mccarthy-tech.com).

It’s not really so many pages, and by passing them out, you won’t have to explain, defend, or evangelize The Core. It’s really just software. Pass it out, and people can take it from there.

1. *Software for Your Head: Core Protocols for Creating and Maintaining Shared Vision*, by Jim and Michele McCarthy. Boston: Addison-Wesley, 2002 (ISBN: 0-201-60456-6).

We offer courses, BootCamps, and certification services for practitioners of The Core protocols, services that are detailed at www.mccarthy-tech.com. Nothing in this distribution authorizes you to actually teach The Core protocols, or certifies you as competent to do so. You are not authorized to conduct a BootCamp on The Core. If you are participating in some type of authorized course or BootCamp, your instructor will have shown you his or her certification from McCarthy Technologies by now, which authorizes the activity. If this hasn't happened, it is an unauthorized activity.

Incidentally, if you have already heard about The Core protocols from someone who purportedly knew all about them, but you've never seen this note, now you know something new about that person.

Thank you for your interest in our work.²

*Jim McCarthy
jim@mccarthy.net*

*Michele McCarthy
michele@mccarthy.net*

2. You can find current versions of these pages plus book and video information at www.mccarthy-tech.com.

THE CORE PROTOCOLS

THE CHECK IN PROTOCOL

The Specific “In-ness” Commitments

When you say, “I’m in” (see “Group Check In”), you commit to the following behaviors:

- You will listen and observe fully.
- You will offer to the team and accept from the team only rational, efficient behavior.
- If the team or its members stray from the CheckIn commitments, you will mention the deviation as soon as you are aware of it and recommend alternative action. If disagreement about your perception arises, you will efficiently propose appropriate alternative action and resolve the conflict using Decider.
- You will accept explicit emotional information as valuable.
- You will be aware of your ratio of time spent effectively speaking to your time spent listening.
- You will speak only and always when you
 - Have a relevant question.
 - Require more information about the current idea. In that case, you will frame requests for information succinctly and clearly.
 - You will ask no bogus questions—that is, questions that reveal your opinions rather than investigate another’s thinking. An example of a good question: “Jasper, will you say more about [whatever]?”
 - Have a relevant proposal.
 - Have an official speaking role in a Decider.
 - Have immediate, relevant value to add.
 - Are responding to a request for information.
 - Are volunteering a supportive idea to the current speaker. You will ask the speaker if he or she wants your idea before stat-

ing it. The current speaker, of course, is free to accept, investigate, or reject your offer.

- Are performing a CheckOut or a CheckIn.
- Express a better idea than the current one (idea preamble).
In exchange for the opportunity to present your idea, you commit to uphold your idea until one of the following is true: (1) Your idea is shown to you to be unsuitable or inefficacious; (2) your idea is expanded in a way that includes or transcends its original value; or (3) your idea is resolved in a Decider process.

Personal Check In

Anyone on the team can check in as, when, and if he or she desires. No permission is required to do so. In the case of a personal CheckIn, no participation beyond listening is required from other team members. When you want to check in, you say, “I’m going to check in.” This activity takes precedence over any other Core activity except running a Decider session.

Group Check In

Although the purpose of the CheckIn protocol is to facilitate the engagement of the person who checks in, it is more efficient if a general group CheckIn takes place. This situation brings the requirement that every team member will check in or pass.

Usually, a group CheckIn takes place at the beginning of a meeting or other team gathering, after a break in a long team meeting, or when the group’s activities or direction is confusing or conflict-laden. Group CheckIn also occurs at the beginning of telephone meetings, in any contact between individuals, or in electronic chats. To inaugurate a group CheckIn, simply suggest, “Let’s check in.” You, as the invoker of a group CheckIn, must check in first.

Execution of a group CheckIn proceeds as follows:

1. Start with the invoker. Each person takes a turn when he feels it is appropriate until everyone is “in” or has “passed.”

2. Each person says, “I pass,” or “I feel (sad and/or mad, and/or glad, and/or afraid).” (Optionally, each person might give a brief explanation of emotional state.)
3. Say, “I’m in.” This statement seals your commitment as outlined in the CheckIn commitments.
4. The group responds, “Welcome.” This statement acknowledges that they heard your check-in and accept your commitment to be “in.”

Example

Person Checking In: “I feel afraid and glad and sad. I feel afraid that this new project won’t be exciting or that it won’t turn out well. But I feel glad that we are starting a new project. Also, I feel sad that I’m not with my family today. And I’m in.”

Group: “Welcome.”

Core Emotional States

CheckIn requires that all feelings be expressed in terms of four and only four emotional states:

- Mad
- Sad
- Glad
- Afraid

Although myriad other emotions exist, all can be expressed with acceptable fidelity in terms of mad, sad, glad, and afraid.

It is not legal to check in without referencing any of the four emotional states unless you pass. Also, it is not legal to introduce emotions other than those in the four emotional states.

When to Use Check In

- Check in at the beginning of any meeting. You can check in individually or call for a general CheckIn. If you call for a gen-

eral CheckIn, you cannot pass and you must be the first person to check in.

- Check in when the team seems to you to be moving toward unproductive behavior.
- Check in whenever you feel the need.

Check In Guidelines

- CheckIn creates maximal results if you express at least two feelings when checking in.
- Do not describe yourself as “a *little* mad/sad/afraid” or use other qualifiers that diminish the importance of your feelings.
- Check in as deeply as possible (where “depth” can be thought of as the “degree of disclosure and extent of feelings of vulnerability that result”); the depth of a team’s CheckIn translates directly to the quality of the team’s results.

Check In Rules

CheckIn is a time apart, and is governed by these constraints:

- No discussion is allowed during CheckIn—only the welcome at the end of each CheckIn. Simply listen to each person, speak when it is your turn, and wait until everyone is done before speaking additionally. Listen and observe as deeply as possible. Gauge and note the congruency of your own emotional response to the CheckIn.
- Do not talk about your own CheckIn before or after CheckIn.
- Do not ask about, reference, or disclose another’s CheckIn.
- Do not blame others for your emotions during CheckIn.
- Do not use CheckIn to talk about, yell at, get “pissed off” at, or confront another team member or anyone not present.

THE CHECK OUT PROTOCOL

Say, “I’m checking out.” Then immediately physically leave the group until you’re ready to check in once again.

Synopsis

Use your time off in a way that will allow you to return refreshed and participate fully, even though you may not feel like using your time in that way. While there is no shame in checking out, your lack of contribution should inspire concern. Give it some thought. Also, you may experience discomfort when you check out, but it will soon be lost in the greater sense of relief that you feel in living out your commitments.

Others may become alarmed by what they perceive as your abrupt disconnection, but they'll survive any momentary discomfort and even prosper in your absence. Avoid the temptation to exploit their potential for alarm by making a show of your Check Out (e.g., by dramatizing the viewpoint that *they* are driving you to check out).

When to Use Check Out

When you need time to take care of yourself in any way (e.g., calm down, rest, or do what is necessary to return fully checked in). CheckOut gives you and your team the opportunity to be productive simultaneously when that is impossible if you remain.

CheckOut is also used when individuals need to take care of personal matters.

Check Out Commitments

- To admit your lack of productive engagement and physically leave
- To not check out to get attention
- To return as soon as you can be productively engaged again
- To return without unduly calling attention to your return
- To be clear with the team about your checking out. (For instance, tell the entire team when you are checking out, not just one person. If you are checking out for more than an hour or so, let your teammates know when you will return.)

Check Out Guidelines

CheckOut is an admission that you are unable to contribute at the present time. It is intended to help the team, not to manipulate team members. This pattern is not intended for any of the following purposes:

- To express your anger
- To cause disruption
- To draw attention
- To create drama
- To trigger others' feelings

You can tell when it's time to check out if the idea occurs to you.

THE PASSER PROTOCOL

At an appropriate time (presumably at the beginning of some process or protocol), say, "I pass." If you know you will pass on something, you are obliged to do so as soon as you are aware of your decision. Once something is started, you can still pass.

Example

A CheckIn is occurring. You don't want to check in, so at an appropriate point (earlier is better) during the process, you signal the group by saying, "I pass. I'm in."

Synopsis

Passing expresses your decision not to participate in an event—that is, to opt out of a process. Passing sets a margin of safety for everyone. It takes courage.

Passing Guidelines

- Passing is always permissible except during a Decider vote.
- There is no discussion about a person's passing.
- To invoke your right to pass, you must say, "I pass." Silent passing is not allowed. Silence indicates that you are awaiting your turn.

- Inevitably others will be curious. Do not explain your passing.
- You can “unpass.”

When to Use Passer

Do it when and if desired—even if you just want to see how it feels to pass.

Passer Commitments

The following commitments are required with Passer:

- To take good care of yourself
- To not judge, shame, hassle, or interrogate anyone who passes
- To not judge, shame, hassle, or interrogate those who do not pass
- To not explain why you are passing (no matter how great the urge)

If you feel the need to “punish” the group, or you desire to use passing for some other reason than simply wanting to opt out of some activity, something more is likely afoot. You probably need to check out rather than pass. Like CheckOut, the Passer protocol should not be used for dramatic purposes. A temporary inclination to dramatics is always a good reason to check out.

THE DECIDER PROTOCOL

1. The **proposer** says, “I propose . . .”.
2. The proposer offers a concise, actionable proposal.
 - No more than one issue is resolved per proposal.
 - The behavior expected of the voters if the proposal is accepted is clearly specified.
3. The proposer says, “1-2-3.”
4. All team members vote simultaneously in one of three ways:
 - “Yes” voters raise their arms or give a thumbs-up.

- “No” voters point their arms down or give a thumbs-down.
 - “Support-it” voters raise their arms midway or show a hand flat.
5. Once the vote is taken, use the Decider tally procedure:
- If the combination of “no” voters (called *outliers*) and “support-it” voters is too great (approximately 30 percent or more, as determined by the proposer), the proposer drops the proposal.
 - If any of the “no” voters states his absolute opposition to the proposal, the proposal is dead.
 - If there are just a few “no” voters, the proposer uses the Resolution protocol to resolve things with the outliers (the “no” voters).
 - Otherwise, the proposal passes.

Guidelines

1. The proposer is responsible for tallying.
2. No one speaks during Decider except the proposer
 - When stating the proposal or
 - When using ResolutionOr the “no” voter
 - When using Resolution or
 - When declaring his absolute “no” state.
3. “Yes” or “support-it” voters cannot speak during Resolution.
4. Voters requiring more information must vote “no” to stop the proposal before seeking information.
5. Voters do not state why they voted as they did.
6. What constitutes “too many” of a given category of votes (for example, too many “no” votes or too many “no” votes plus “support-it” votes) is determined solely by the proposer. Typically, three or four “no” votes out of ten total votes are considered “too

- many” to pursue to Resolution. A majority of “support-it” votes suggests a very weak proposal.
7. Passing is not allowed on a Decider proposal. You must vote if you are present.
 8. Unanimous “yes” votes or “yes” votes mixed with some “support-it” votes are the only configurations that cause a proposal to be adopted as a part of the team’s **plan of record**.
 9. Each team member is accountable for personally carrying out behaviors specified in a Decider decision, and no member has more or less accountability than any other. Each is also accountable for insisting that the behavior specified in the proposal is carried out by the other team members.
 10. After a proposal passes, a team member who was not present during the vote is responsible for acquiring information about what transpired, and will also be held accountable for the decision. If the person prefers not to be accountable (that is, he would have voted “no” if present), he now must make a new proposal as soon as possible. In the meantime, the individual is bound by the decision just as if he had voted “yes.”
 11. When a “no” voter states that he “won’t get in no matter what” (i.e., an “absolute no” vote), it means that there is no condition that the voter can imagine that would change his vote.
 12. It is traditional, though not mandatory, for an “absolute no” voter to make a new proposal following the death of the proposal killed with his vote.

Voting

Given a proposal, the Decider protocol provides three possible voting strategies:

- Yes
- No
- Support-it

“Support-it” is a “yes” vote with an attitude. It can be translated as, “I can live with this proposal. I believe that it is probably the best

way for us to proceed now. I support it, even though I have some reservations. While I don't believe I can lead the implementation of this proposal, I do commit not to sabotage it."

Decider Outcomes

Three outcomes are possible:

- *Affirmative decision.* Immediate and universal acceptance of the proposal occurs.
- *Efficient negotiation with conflicts exposed and the proposal resolved.* Finer proposals are created while the team's inclusion effort proceeds.
- *Swift elimination of unsupported ideas.* Immediate, clear, and unremorseful rejection of an idea too many people think misguided occurs.

Decider Commitments

Decider requires the following commitments from team members:

- Actively support the decisions reached, with the behavior specified in them.
- Vote your true beliefs.
- Speak or don't speak as specified above.
- Hold others accountable for their decisions.
- Respect an "absolute no" voter. Do not pursue the voter or analyze his motives.
- Do not collect others' votes before making your own.
- Do not repeat failed proposals unless relevant circumstances have changed.
- Keep informed about Decider votes run in your absence and resolve, via Decider, your lack of support, if any, for the decisions made when you were absent.
- Reveal immediately whether you are an "absolute no" voter when you vote "no."

THE RESOLUTION PROTOCOL

When a Decider vote yields a small minority of outliers, the proposer quickly leads the team, in a highly structured fashion, to deal with the outliers. The proposer's goal is straightforward and unabashedly promotes the proposal: to bring the outliers in at least cost.

1. The proposer asks each outlier to express his requirements for joining the team in support of the proposal: "What will it take to bring you in?"
2. The outlier has only two possible legal responses:
 - He may state, at any time after the vote, but no later than when asked the above question by the proposer, that there is "no way" he will change his vote to "yes" or "support-it." This simple declaration means that the proposal is now officially dead, and the Decider protocol ends.
 - The outlier may state in a single, short, declarative sentence precisely what it is he requires to be "in." In this way, he expresses a contingent commitment to see that the proposal is accepted and transformed into reality. If given what he requires, the outlier promises to drop all resistance to the proposal and to provide affirmation and support for it instead.
3. As needed and as possible, the proposer makes an offer to the outlier. Two methods for incorporating changes into the original proposal while resolving any resulting perturbations to non-outliers' support are permitted:
 - If the adaptations to the proposal to accommodate the outlier's requirements are minor, the proposer may employ a simple, unofficial "eye-check" of the nonoutliers to see if there is general acceptance to the changed proposal. If you are opposed to this implicit new proposal, or you require a formal restatement and a new vote, you must make your requirement known during this interval.
 - If the required changes are more complex, the proposer must create and submit a new proposal that accounts for the out-

lier's requirements. The team reviews this proposal and conducts a new vote, and the Decider protocol begins anew.

4. "Yes" voters and "support-it" voters are not allowed to speak during Resolution.
5. If all outliers change their votes from "no" to "support-it" or "yes," then the decision to adopt the proposal is committed; it will be acted upon by the team. No further communication is required to achieve strong, unanimous consensus.

Resolution Commitments

You are committed to answer the question, "What will it take to get you in?", with an actionable modification of the proposal. Explaining why you don't like the proposal, for example, is off the subject.

When to Use Resolution

Use Resolution whenever a small percentage of the team votes "no" to your Decider proposal.

THE INTENTION CHECK PROTOCOL

IntentionCheck assesses the integrity of your own (and, to a lesser extent, another's) intention. IntentionCheck evaluates conditions that tend to skew or bias your effectiveness in dealing with a given issue at the time you run the check.

1. Ask yourself, "Is my current emotional state solid, turbulent, or intense?"
2. Ask yourself, "Is my current receptivity to new information high, medium, or low?"
3. Ask yourself, "Do I understand clearly what my current purpose is?" That is, what result do I want?
4. If your emotional state is not solid, and/or your receptivity is not good, or you are not clear about your purpose, then postpone your action, or use the CheckOut protocol to get the information needed to clear up your intention before acting or speaking.

Synopsis

Checking your intention prior to significant behaviors will improve the odds that your behavior will have the desired results. The most common problem in being effective is low quality of intention. By invoking an `IntentionCheck` on yourself or inviting investigation of your intention, you will act less with more results. This is the essence of efficiency.

When to Use Intention Check

- When ambiguity or uncertainty surrounds your motive
- When your behavior seems likely to discomfort others
- When your behavior will slow others in achieving their goals
- When you are contemplating an interpersonally risky or ethically complex endeavor
- After you failed to use `IntentionCheck`
- When you have strong feelings about another person's behavior, are involved in it to some extent, and are about to engage with him
- When you are psychologically attached to a particular goal and believe others to be resistant to it

THE ALIGNMENT PROTOCOL

The Alignment protocol governs behavior during alignment. It has five major steps and calls upon the `PersonalAlignment`, `PerfectionGame`, and `Investigate` protocols.

1. Team members begin to align themselves by applying the `PersonalAlignment` protocol.

All alignees start out with the same, default `PersonalAlignment` statement: *"I want self-awareness."*

Spend sufficient private time in introspection and in receiving help, so you will be able to form and express your `PersonalAlignment` to the team.

Small groups of people can gather to listen to and investigate one another, but only after all of the individuals have spent time alone and in receiving help.

2. Team members use Investigate to help one another complete their PersonalAlignments.

Investigate is used by team members who are not currently working on their own Alignment, when they make themselves available to an alignee during the public portion of his PersonalAlignment.

3. Alignees iterate as necessary.

Your first pass at PersonalAlignment, with the support of the team in Investigate mode, will generally produce a completed PersonalAlignment, but it may yield an inconclusive result or provoke a change of heart. It may lead to a deepening of the Alignment or just confusion. It is acceptable to redo the public portion of a PersonalAlignment. Even if everything about your Alignment seemed clear, if you desire to change some or all of it, you may do so—provided that you share the changes with the team in full meeting.

4. The team improves the collected PersonalAlignment statements.
5. The team completes a WebofCommitment and integrates it into the group.

Additional factors contribute to the quality and ease of Alignment, including these physical issues.

- Alignment is best achieved in a quiet, comfortable place, away from usual workplaces and workplace stresses.
- The team will maximize its effectiveness if it performs all PersonalAlignments as a team during a single, off-site session, taking as much time as necessary (usually several days) to do the work to everyone's satisfaction.
- If one or more persons on the team can't attend the off-site meeting, the team must proceed without them.

THE PERSONAL ALIGNMENT PROTOCOL

Complete the following PersonalAlignment exercise. You are encouraged to ask others for help early and often. Expect your helpers to use the Investigate pattern, and when you help others with their PersonalAlignment, you must use the Investigate pattern.

1. Ask yourself, “What do I want? What, specifically, do I, personally, want?”
2. When you think you know what you want, write it down.
3. Now ask yourself, “Why don’t I have what I say I want already?” Assume that you could have had it by now. Almost always, there is some internal blocking element preventing you from getting it, or you already would have it. Write down your answer.

If your answer to the question in step 3 blames or defers accountability to uncontrollable circumstances or other people, pretend your last answer to step 3 is just a story, a myth that somehow deprives you of your full power to achieve for yourself. Before proceeding further with this protocol, you must make an imaginative leap to a more personally powerful stance. Likely, you will have to increase your self-awareness. Increase your perception and your receptivity. In any case, change *something* now about the way you have executed this protocol so far, because it hasn’t worked. Then go back to step 3.

4. If your answer to the question in step 3 is more than a few words, simplify your answer, going back to step 3.
5. If your answer to step 3 doesn’t refer to a personal issue:
 - Increase your commitment to yourself in this process.
 - Consider whether you are afraid and, if so, what you fear.
 - Consider whether your answer to step 3 shows integrity.
 - Change *something* now about the way you have executed this protocol so far, because it hasn’t worked. Then go back to step 3.

6. If you have gone back to step 3 several times:
 - Employ AskforHelp (again, if necessary).
 - And/or take a break and go back to step 1.
7. If your answer to step 3 points to a problem or constraint that, if solved, would radically increase your effectiveness in life—work *and* play—you have identified a *block*.
8. Until you are certain that what you say you want is what you really want, remain at this point. If you have remained here for a while, you are still uncertain, and your team is moving on, adopt the default alignment: that is, you want more “self-awareness” and you don’t know what’s blocking you. Go to step 13.
9. If you are not certain that eliminating the block identified by your answer to step 3 will be worth a great deal of effort, go back to step 8.
10. Check out the block with people who know you and with people who know about blocks, if possible. If you are unwilling to utilize AskForHelp from your team, go back to step 8.
11. Determine what virtue would be powerful enough to shatter the block.
12. Decide whether this virtue is what you really want: the power that would yield what you *thought* you wanted (in step 1). If it is, write it down. Go to step 3.
13. Create a very concise sentence that begins with the words: “I want . . .”
14. If your sentence has unneeded words, go back to step 13.
15. This sentence is your PersonalAlignment statement. Check it out with all of your team.
16. Ask them if they can think of a shorter, more direct way to say the same thing.
17. Promise them to take specific, visible actions that will show your commitment to obtaining what you want. Tell them what they can expect to see you doing, commencing now.

18. Ask your team for help. Will they do *X*, when you signal them by doing *Y*? In your request for help, there should be specific actions you are asking them to do that will help you obtain what you want. It is very important that *you* initiate this action-reaction sequence by signaling to them that you are working on your PersonalAlignment. It is not up to your teammates to initiate status checks or police your PersonalAlignment. Ask for your teammate's help using very specific language, such as "Bill," you ask, "when I [*do something positive that demonstrates my commitment to attaining what I say I want*], will you [*show a sign of support, encouragement, and/or proffer any requested substantive help*]?"
19. Write or rewrite the following:
- Your personal alignment statement
 - Alignment evidence
 - Support you ask for from your team

Examples of support include the following:

"When I say, 'This takes courage for me,' will you applaud?"

"When I give a daily report on how I took care of myself, will you do the wave?"

"Will you meet with me for one-half hour weekly, Bill, so I can work on this with you?"

Personal Alignment Commitments

This protocol requires the following commitments from the alignee:

- To pass early if you are going to pass. Pass later only if you fail to pass early.
- Move to the deepest desirable point in the shortest possible time.
- Be truthful.
- Be receptive to the effective assistance of others.
- Reject assistance that impedes your progress.

- Don't just "go along," or merely humor the Alignment process. If you are inclined to do that, pass. This choice preserves the integrity of the experience for others.
- "Pretend" as needed. That is, try out new ideas about yourself before discarding them.
- Be accountable.
- Avoid storytelling.
- Insist that when you give your support signal, the team members follow through with their support.

The Personal Alignment Statement

A PersonalAlignment statement begins with the words, "I want." The most common and successful alignment statements have the form

"I want X", or "I want to X."

where *X* is the virtue or power that you have decided will break through your biggest block.

Common Alignments

The following are the most common alignments. This list is a partial list, not meant to serve as a constraint or a boundary.

I want:

Faith	To love myself
Hope	To value myself
Passion	To feel my feelings
Self-awareness	To believe in myself
Self-care	Integrity
Courage	Fun
Wisdom	Ease
Peace	To accept myself
Maturity	To be honest with myself
Presence	To be patient with myself
Joy	

Faux Alignments

Although it is difficult to condemn a whole word or phrase to the trash heap as a faux Alignment, we have found that these “alignments” are always indicators of something amiss in the person’s understanding either of PersonalAlignment or of the phrase in question.

I want:

Confidence	To retire
Self-confidence	To be the best _____
Self-control	To not _____
Strength	Sanity
To solve problems	Knowledge
To listen	Focus
To be understood	Balance
To understand	Patience
Fame	Security
To be rich	

THE INVESTIGATE PROTOCOL

1. Become a detached but fascinated inquirer.
2. Ask only questions that will increase your understanding. Ask questions to acquire information. Maintain the posture of an interested person, handicapped by ignorance. (See “Intention Check” in Chapter Six.)
3. Don’t ask inappropriate questions. For example, avoid the following types of inquiries:
 - Questions that attempt to lead the alignee or that reflect your agenda. This problem can arise when you have strong feelings about the subject.
 - Questions that attempt to hide an answer you believe is true.
 - Poorly thought out questions. If you are not aware of your own intention before you ask the question, don’t ask it. (See the IntentionCheck protocol.)

- Questions that invite the alignee to wander off into too much analysis or irrelevant material. Questions that begin with “Why” can spur this problem.
4. Use a few formulations for your questions. Consider using the following forms:
 - “What about *X* makes *YZ*?” For example, “What about your coding makes the experience frustrating?”
 - “How does it go when that happens?” “Will you slow down the process and describe it to me?” “Take a specific example and slow it down.”
 5. Ask questions only if the alignee is engaged and appears ready to learn more. If your teammate seems to be bored, stubborn, resistant, or going in circles, then stop investigating. The alignee must adhere to the commitments in personal alignment if you are to continue to any good effect. To break up this block, say, “I have a sense that I am pulling information out of you against your will. Let’s take some time to think about this issue and talk about it later.” You can also just be quiet.
 6. Give opinions rarely and only after receiving the alignee’s permission. Stick to your intention of gathering more information. If you have an interesting thought, a good idea, or theory, say, “I have an [. . .]. Would you like to hear it?” The alignee can then answer “yes” or “no” or state conditions under which your input would be welcome.

If you feel that you will explode if you can’t say what’s on your mind, that’s a good indication that you shouldn’t speak.
 7. Never argue during PersonalAlignment.
 8. Don’t talk to anyone other than the alignee.

When to Use Investigate

- When an alignee asks for your help with a PersonalAlignment.
- In an Alignment situation, when an alignee has explicitly stated that Investigate questions are welcome

- In general, when you are learning about a phenomenon, with an eye toward exploiting it
- When you are working on your own PersonalAlignment.

Investigate Commitments

- Intensify your curiosity.
- Widen your receptivity.
- Ask well-formed questions.
- Set aside your biases toward and prior experiences with the alignee. Observe the alignee with innocence and a fresh perception.
- Accept what the alignee says while perceiving more than usual.
- Do not tolerate theorizing about the alignee.
- Do not tolerate diagnosis of the alignee.
- Do not tolerate therapy during Alignment.
- Do not tolerate any distraction away from the alignee.
- Use Investigate or CheckOut.

Examples of Investigative Questions

What is the one thing you want most from this project?

What blocks you from getting what you want?

If that block were removed, would you get what you want?

Is there some virtue that would enable you to eliminate the block?

What is the biggest problem you see?

What is the most important thing you could do right now?

If you could have anything in the world right now, what would it be?

If you could do anything in the world right now, what would it be?

How does it go when that happens?

Would you explain a specific example?
Would you slow it down into steps?

THE WEB OF COMMITMENT PROTOCOL

The WebofCommitment protocol has four steps:

1. Each alignee should create a list that includes the following:
 - A PersonalAlignment statement
 - Positive, measurable evidence—both short-term and long-term—that will show you are getting what you want
 - Support commitments from your team in the form of (1) a specific positive signal you give to your team and (2) a specific positive show of support your teammates give to you
2. Post the list in a public place—on a bulletin board, as a poster, or in an e-mail.
3. Conduct a ceremony for the entire team to do the following:
 - Highlight each PersonalAlignment
 - Bring the PersonalAlignment process to a close (optional)
 - Celebrate the team alignment
4. Keep your commitments to one another, and track whether commitments are kept. Renew all elements as needed.

THE ASK FOR HELP PROTOCOL

The AskforHelp protocol involves two roles: an asker and a helper.

Asker Role

When you are the asker, you must inaugurate the help transaction, as follows:

1. State some form of the following question to your intended helpers: “[Name of the person you are soliciting], will you

help me [verb] [object being created, goal being reached, and so on] . . . ?”

2. If you have a specific activity or activities you desire from the helper, and especially if these are the only activities you are willing to accept, express these specifics before encouraging the would-be helper to answer your request.
3. You must always shape your help request (as in steps 1 and 2) so that you ask a question that begins “Will you . . . ?”
4. After asking for help with a Core-legal question, say nothing until your question is answered.

Helper Role

When addressed directly and properly with a request for help, engage your full attention on the asker. There are only four legal responses to a valid AskforHelp request:

1. If you are unable to fully engage with the asker on the request for help, immediately say, “[Name of asker], I can’t discuss this request right now.” Then, if possible, arrange a mutually convenient time to discuss the issue.
2. If, after focusing your attention on the asker and listening to the request, you don’t want to carry out (or even further discuss) the request, tell the asker, “No, I won’t do that,” or simply “No.” Then say nothing else.
3. If you are willing to help with or willing to discuss the request, but need more information about the request, its purpose, or any specifics, ask, “Will you tell me more about the specifics of what you require?” You can then ask questions about the request to get the information you need. Once you understand the specifics, then answer “yes” or “no,” which ends the protocol. Otherwise, go to step 4.
4. If you want to offer help, but believe the help requested is not what you can or should do, decline the request explicitly before

proceeding further. Answer something like “No, I won’t. But I will [state the thing you think would be more helpful]. Would that be helpful to you?”

Asker Commitments

The asker should commit to the following:

1. Have a clear intention. A person who is aware of his desire for help may often misstate this intention to secure help and somehow induce in the helper the urge to rescue the asker. (A rescue occurs when help is offered but not explicitly requested.)
Examples of breaking protocol include the following:
 - “I could use a little help.”
 - “I need . . .”
 - “If I had some help . . .”
 - “I want help here.”
 - “Help!”
2. Be utterly clear, in your own mind and in your request, that you are the asker—the supplicant in the help transaction. This recognition is important to the helper because your asking must be freely offered, the helper must perceive that you know that he can decline the request, and the transaction must carry no penalty to the helper if he does decline.
3. State the specifics, if any, of your request.
4. Assume that the person from whom you’re requesting help accepts the responsibility to say “no.” That is, don’t excuse your failure to ask for help by claiming responsibility for determining others’ limits.
5. Don’t apologize or otherwise obscure your intention.
6. Accept “no” without any additional internal or external emotional drama.
7. Accept the help offered as completely as possible. If you don’t understand the value of what is offered, feel that it wouldn’t be

useful, or believe yourself to have considered and rejected the idea offered previously, assume a curious stance instead of executing a knee-jerk, “But . . .” rejection.

8. Ask for something positive.
9. Accept genuine help.

Helper Commitments

The helper should commit to the following:

1. To say “no” when you don’t want to help, or even when you aren’t sure you want to help
2. To say you have changed your mind and don’t want to help if you begin to help and decide that you really don’t want to do so
3. To fulfill completely any of your commitments to help
4. To say “no” without drama or rancor or soliciting approval from the asker
5. To offer what you believe is truly helpful if you have something that you believe would be useful to the asker, even if it is not exactly what he originally requested.

SHARED VISION PROTOCOL

1. Envision a vision-driven life. Turn that image into a metavision (a vision of how you create and use visions) with and for your team and your institution.
2. As a team, decide what kind of world you will create. Answer two questions together: (1) How will the world be different when you finish your work? (2) What will life be like for you and your customers? From your answers create a FarVision, and write a FarVision statement.
3. Deliver one version of your product after another, and deliver each version on time. Each version must be designed to validate the single, the one and only, message you promulgate with it. The product must also palpably contribute to the increasing legitimacy of your FarVision. That is, each version must demon-

strably bring you and your customers closer to the world of your FarVision.

THE FAR VISION PROTOCOL

413

After having completed Alignment, with your team, write a statement that best expresses for all of you what the world will look like when your work together is done.

- *The FarVision must be imaginative.* Look as far into the future as possible. Twenty years is a good starting point, but the date chosen must be always beyond your ability to extrapolate current trends. That is, it must be the work of intention and imagination, not analysis.
- *The FarVision must be measurable.* Ideally, progress can be measured as well. The desired result may be an observable, external thing or event, such as “put a man on the moon.” Alternatively, it might be softer and more difficult to measure, such as “create infinite, free bandwidth.” Your FarVision could also be values-driven, which is more difficult but still possible to measure, such as “eliminate poverty” or “create ubiquitous radical democracy.”
- *The FarVision statement should just be a few words,* ideally no more than ten words. If it is more than six words, ask your team to reevaluate it.
- *Use the PerfectionGame to perfect your FarVision.*
- *The team should unanimously support the FarVision,* using Decider.

Examples of FarVision statements and version statements follow:

Put a man on the moon.

Version 1: Orbit the earth.

A computer on every desk.

Version 1: Software that’s easy to use.

World peace.

Version 1: Peace in our country.

THE PASSIONOMETER PROTOCOL

The Passionometer protocol provides a straightforward technique for discovering what a team cares about and how much a team cares about it. Its most common application is the creation of the team's vision statement.

1. On index cards team members write down meaningful words or phrases that they associate with the world of the team's FarVision.
2. Toss the cards onto the floor faceup. If repeats are found, throw out the extras or keep them together as a set.
3. A facilitator holds up each card (or set), and team members show their passion for the word or phrase by making noise or vivid gestures, or by another means.
4. The facilitator, with the advice and consent of the team, makes three piles of the cards based on the responses from the team as he flips through the cards one at a time:
 - Highly passionately supported
 - Somewhat passionately supported
 - No real passion
5. Repeat steps 3 and 4, using only those cards that are highly passionately supported until the cards are narrowed down to a set of five or six.
6. If the team remains stuck, repeat Passionometer on the highly passionately supported cards.

THE PERFECTION GAME PROTOCOL

1. Players sit in a circle.
2. Each person in the circle names a task that he believes to be simple and that the individual is willing to perform throughout the game—for example, “snapping my fingers,” “whistling a short tune,” or “acting dead.”

3. The first player performs the task named in step 2. This performance has the following structure:
 - The player alerts the rest of the group to the beginning of the performance by saying, “Okay, I’m starting now.” Everything the player does after this point is subject to perfecting.
 - The player performs his task.
 - The player says, “I’m done.” Everything up to but not including this statement is subject to perfecting.
4. The remaining players rate the player’s performance on a scale of 1 to 10, where 10 is a perfect performance of the task. The rating must be supported with critical analysis of a particular form: After saying the score (for example, “I give your performance a 7”), the scorer must state the following:
 - Specifically, what about the performance was good and what earned the points in the score
 - Specifically, what the performer must do in the next iteration of the performance to be awarded a perfect 10

The next player then performs his task and is rated by the rest of the group as described above.

5. Steps 1–4 are completed two more times, so that each player performs and is rated three times. Each person plays the role of critic for the rest of the team members in between each of his own performances.

Analysis of the Perfection Game Protocol

Purely or partially negative feedback is not allowed at any point during the PerfectionGame protocol. For example, “I don’t like the sound of the finger snap.” The important information to transmit in this case may be something like, “The ideal sound of a finger snap for me is one that is crisp, has sufficient volume, and startles me somewhat. To get a 10, you would have to increase your crispness.”

If you cannot think of a better alternative performance, you cannot withhold points. The default score is a perfect 10.

You must follow the scoring routine exactly:

- “I rate your performance n .”
- “What I liked about it was p, q, \dots, z .”
- “What it would take to get a 10 from me is a, b, \dots, z .”

If one person breaks the protocol, the other team members must politely correct the offending person by pointing out the infraction. They must then remind the offender of the correct protocol immediately by suggesting, “I give it a” “What I liked about it was,” or “What it would take to get a 10 is,” as appropriate.

When playing the PerfectionGame, the team will develop a sense of the ideal performance of any given act. This aesthetic will take into account the best suggestions made, with lesser suggestions being abandoned.

Including each of the suggested improvements into the next performance rarely yields a perfect performance. The “perfecters” could be wrong about their prior feedback (not intentionally, of course), or the combination of all suggestions may have a negative effect on the performance. As the “perfectee,” you must accept only the superior criticism of your performance and implicitly reject the inferior feedback.

Your ratings must not use a “dislike” to “like” scale, where 1 is “completely dislike” and 10 is “completely like.” The perfection game is not about whether you “like” something. The rating scale goes from 1, “The thing has no value now and I can add all value needed in my feedback,” to 10, “The thing has full value and/or I can’t think of anything that would make it better.” It is important to hold perfecters accountable to this type of scale and respectfully correct them if you see the dislike/like scale coming into play.

In addition, the rating must be reasonable. For instance, if you rate a performance as an 8, you are saying that it is 80 percent perfect and/or you can tell the person exactly how to gain the 20 percent of missing value. You must not give an 8 and then provide only 1 percent of the missing value.

The “what it will take to get a 10” portion of the game may not be performed in writing. It must be performed verbally with the perfectee.

If you feel an impulse to grade on the dislike/like scale, can’t give a reasonable amount of value that correlates with your rating and are unwilling to raise your rating accordingly, or feel the need to write your perfecting down instead of speaking to the person, then you should pass. These impulses can contribute to a negative feedback cycle that distracts the team from achieving the desired results.

L I C E N S E A G R E E M E N T

The Core Protocols V. 1.0

December 2001

Jim and Michele McCarthy, www.mccarthy-tech.com

Copyright © 2001 McCarthy Technologies, Inc.

Permission is granted to make and distribute verbatim copies of this software provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this software under the conditions for verbatim copying, provided also that the section entitled “GNU General Public License” is included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through

that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE

PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

423

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

One line to give the program’s name and an idea of what it does

Copyright © yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © year, name of author, Gnomovision comes with ABSOLUTELY NO WARRANTY; for details, type “show w”. This is free software, and you are welcome to redistribute it under certain conditions; type “show c” for details.

The hypothetical commands “show w” and “show c” should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than “show w” and “show c”; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program “Gnomovision” (which makes passes at compilers) written by James Hacker.

Signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.