

Purpose

The standard stock of actions supplied with Inform, along with the rules which define them; and the Understand grammar which corresponds to them.

A/sr4. §2-5 Taking inventory; §6-9 Taking; §10-11 Removing it from; §12-15 Dropping; §16-19 Putting it on; §20-23 Inserting it into; §24-27 Eating; §28-31 Going; §32-35 Entering; §36-39 Exiting; §40-43 Getting off; §44-46 Looking; §47-49 Examining; §50-52 Looking under; §53-55 Searching; §56-57 Consulting it about; §58-61 Locking it with; §62-65 Unlocking it with; §66-69 Switching on; §70-73 Switching off; §74-77 Opening; §78-81 Closing; §82-85 Wearing; §86-89 Taking off; §90-93 Giving it to; §94-95 Showing it to; §96-97 Waking; §98-99 Throwing it at; §100-101 Attacking; §102-103 Kissing; §104-105 Answering it that; §106-108 Telling it about; §109-110 Asking it about; §111-112 Asking it for; §113-114 Waiting; §115-116 Touching; §117-119 Waving; §120-122 Pulling; §123-125 Pushing; §126-128 Turning; §129-130 Pushing it to; §131-133 Squeezing; §134-137 Saying yes; §138-139 Burning; §140-141 Waking up; §142-143 Thinking; §144-145 Smelling; §146-147 Listening to; §148-149 Tasting; §150-151 Cutting; §152-153 Jumping; §154-155 Tying it to; §156-157 Drinking; §158-159 Saying sorry; §160-161 Swearing obscenely; §162-163 Swearing mildly; §164-165 Swinging; §166-167 Rubbing; §168-169 Setting it to; §170-171 Waving hands; §172-173 Buying; §174-175 Singing; §176-177 Climbing; §178-179 Sleeping; §180-185 Out of world actions; §186 Miscellaneous Grammar Tokens; §187 Grammar

§1. NI comes with no actions built in, and only one (perhaps unexpected) assumption: that the 8th action defined is “going”.

The order, and the subheadings, here are responsible for the order and subheadings used in the Actions page of the Index.

Part SR4 - Actions

Section SR4/1 - Generic action patterns

Section SR4/2 - Standard actions concerning the actor's possessions

§2. Taking inventory.

Taking inventory is an action applying to nothing.

The taking inventory action translates into I6 as "Inv".

The specification of the taking inventory action is "Taking an inventory of one's immediate possessions: the things being carried, either directly or in any containers being carried. When the player performs this action, either the inventory listing, or else a special message if nothing is being carried or worn, is printed during the carry out rules: nothing happens at the report stage. The opposite happens for other people performing the action: nothing happens during carry out, but a report such as 'Mr X looks through his possessions.' is produced (provided Mr X is visible)."

§3. There used to be a rule, documented here, to do with pronouns, and this was explained in terms of Misse Lee, a black and white cat living in North Oxford; named for a Cambridge-educated pirate queen in the South China seas who is the heroine – or villainess – of the tenth in Arthur Ransome's *Swallows and Amazons* series of children's books, *Misse Lee* (1941). The rule was then removed, but it seemed sad to delete the only mention of Misse, and all the more so since she died (at a grand old age and in mid-spring) in 2008.

§4. Carry out.

Carry out taking inventory (this is the print empty inventory rule):

```
if the first thing held by the player is nothing, stop the action with
    library message taking inventory action number 1.
```

Carry out taking inventory (this is the print standard inventory rule):

```
issue library message taking inventory action number 2;
say ":[line break]";
list the contents of the player, with newlines, indented, including contents,
    giving inventory information, with extra indentation.
```

§5. Report.

Report an actor taking inventory (this is the report other people taking inventory rule):

```
if the actor is not the player,
    issue actor-based library message taking inventory action number 5 for the actor.
```

§6. Taking.

Taking is an action applying to one thing.
The taking action translates into I6 as "Take".

The specification of the taking action is "The taking action is the only way an action in the Standard Rules can cause something to be carried by an actor. It is very simple in operation (the entire carry out stage consists only of 'now the actor carries the noun') but many checks must be performed before it can be allowed to happen."

§7. Check.

Check an actor taking (this is the can't take yourself rule):

if the actor is the noun, stop the action with library message taking action number 2 for the noun.

Check an actor taking (this is the can't take other people rule):

if the noun is a person, stop the action with library message taking action number 3 for the noun.

Check an actor taking (this is the can't take component parts rule):

if the noun is part of something (called the whole), stop the action with library message taking action number 7 for the whole.

Check an actor taking (this is the can't take people's possessions rule):

let the local ceiling be the common ancestor of the actor with the noun;
let H be the not-counting-parts holder of the noun;
while H is not nothing and H is not the local ceiling:
if H is a person, stop the action with library message taking action number 6 for H;
let H be the not-counting-parts holder of H;

Check an actor taking (this is the can't take items out of play rule):

let H be the noun;
while H is not nothing and H is not a room:
let H be the not-counting-parts holder of H;
if H is nothing, stop the action with library message taking action number 8 for the noun.

Check an actor taking (this is the can't take what you're inside rule):

let the local ceiling be the common ancestor of the actor with the noun;
if the local ceiling is the noun, stop the action with library message taking action number 4 for the noun.

Check an actor taking (this is the can't take what's already taken rule):

if the actor is carrying the noun, stop the action with library message taking action number 5 for the noun;
if the actor is wearing the noun, stop the action with library message taking action number 5 for the noun.

Check an actor taking (this is the can't take scenery rule):

if the noun is scenery, stop the action with library message taking action number 10 for the noun.

Check an actor taking (this is the can only take things rule):

if the noun is not a thing, stop the action with library message taking action number 15 for the noun.

Check an actor taking (this is the can't take what's fixed in place rule):

if the noun is fixed in place, stop the action with library message taking
action number 11 for the noun.

Check an actor taking (this is the use player's holdall to avoid exceeding
carrying capacity rule):

if the number of things carried by the actor is at least the
carrying capacity of the actor:
if the actor is holding a player's holdall (called the current working sack):
let the transferred item be nothing;
repeat with the possible item running through things carried by
the actor:
if the possible item is not lit and the possible item is not
the current working sack, let the transferred item be the possible item;
if the transferred item is not nothing:
issue library message taking action number 13 for the
transferred item and the current working sack;
silently try the actor trying inserting the transferred item
into the current working sack;
if the transferred item is not in the current working sack, stop the action;

Check an actor taking (this is the can't exceed carrying capacity rule):

if the number of things carried by the actor is at least the
carrying capacity of the actor, stop the action with library
message taking action number 12 for the actor.

§8. Carry out.

Carry out an actor taking (this is the standard taking rule):
now the actor carries the noun.

§9. Report.

Report an actor taking (this is the standard report taking rule):
if the actor is the player, issue library message taking action number 1
for the noun;
otherwise issue actor-based library message taking action number 16 for the noun.

§10. Removing it from.

Removing it from is an action applying to two things.

The removing it from action translates into I6 as "Remove".

The specification of the removing it from action is "Removing is not really an action in its own right. Whereas there are many ways to put something down (on the floor, on top of something, inside something else, giving it to somebody else, and so on), Inform has only one way to take something: the taking action. Removing exists only to provide some nicely worded replies to impossible requests, and in all sensible cases is converted into taking. Because of this, it's usually a bad idea to write rules about removing: if you write a rule such as 'Instead of removing the key, ...' then it won't apply if the player simply types TAKE KEY instead. The safe way to do this is to write a rule about taking, which covers all possibilities."

§11. Check.

Check an actor removing something from (this is the can't remove what's not inside rule):

 if the holder of the noun is not the second noun, stop the action with
 library message removing it from action number 2 for the noun.

Check an actor removing something from (this is the can't remove from people rule):

 let the owner be the holder of the noun;
 if the owner is a person:
 if the owner is the actor, convert to the taking off action on the noun;
 stop the action with library message taking action number 6 for the owner.

Check an actor removing something from (this is the convert remove to take rule):

 convert to the taking action on the noun.

The can't take component parts rule is listed before the can't remove what's not inside rule in the check removing it from rules.

§12. Dropping.

Dropping is an action applying to one thing.

The dropping action translates into I6 as "Drop".

The specification of the dropping action is "Dropping is one of five actions by which an actor can get rid of something carried: the others are inserting (into a container), putting (onto a supporter), giving (to someone else) and eating. Dropping means dropping onto the actor's current floor, which is usually the floor of a room - but might be the inside of a box if the actor is also inside that box, and so on.

The can't drop clothes being worn rule silently tries the taking off action on any clothing being dropped: unlisting this rule removes both this behaviour and also the requirement that clothes cannot simply be dropped."

§13. Check.

Check an actor dropping (this is the can't drop yourself rule):

if the noun is the actor, stop the action with library message putting it on action number 4.

Check an actor dropping (this is the can't drop what's already dropped rule):

if the noun is in the holder of the actor, stop the action with library message dropping action number 1 for the noun.

Check an actor dropping (this is the can't drop what's not held rule):

if the actor is carrying the noun, continue the action;
if the actor is wearing the noun, continue the action;
stop the action with library message dropping action number 2 for the noun.

Check an actor dropping (this is the can't drop clothes being worn rule):

if the actor is wearing the noun:
issue library message dropping action number 3 for the noun;
silently try the actor trying taking off the noun;
if the actor is wearing the noun, stop the action;

Check an actor dropping (this is the can't drop if this exceeds carrying capacity rule):

let H be the holder of the actor;
if H is a room, continue the action; [room floors have infinite capacity]
if H provides the property carrying capacity:
if H is a supporter:
if the number of things on H is at least the carrying capacity of H:
if the actor is the player,
issue library message dropping action number 5 for H;
stop the action;
otherwise if H is a container:
if the number of things in H is at least the carrying capacity of H:
if the actor is the player,
issue library message dropping action number 6 for H;
stop the action;

§14. Carry out.

Carry out an actor dropping (this is the standard dropping rule):
now the noun is in the holder of the actor.

§15. Report.

Report an actor dropping (this is the standard report dropping rule):
if the actor is the player, issue library message dropping action number 4
for the noun;
otherwise issue actor-based library message dropping action number 7 for the noun.

§16. Putting it on.

Putting it on is an action applying to two things.

The putting it on action translates into I6 as "PutOn".

The specification of the putting it on action is "By this action, an actor puts something he is holding on top of a supporter: for instance, putting an apple on a table."

§17. Check.

Check an actor putting something on (this is the convert put to drop where possible rule):

if the second noun is down or the actor is on the second noun,
convert to the dropping action on the noun.

Check an actor putting something on (this is the can't put what's not held rule):

if the actor is carrying the noun, continue the action;
if the actor is wearing the noun, continue the action;
issue miscellaneous library message number 26 for the noun;
silently try the actor taking the noun;
if the actor is carrying the noun, continue the action;
stop the action with library message putting it on action number 1 for the noun.

Check an actor putting something on (this is the can't put something on itself rule):

let the noun-CPC be the component parts core of the noun;
let the second-CPC be the component parts core of the second noun;
let the transfer ceiling be the common ancestor of the noun-CPC with the second-CPC;
if the transfer ceiling is the noun-CPC,
stop the action with library message putting it on action number 2 for
the noun.

Check an actor putting something on (this is the can't put onto what's not a supporter rule):

if the second noun is not a supporter,
stop the action with library message putting it on action number 3 for
the second noun.

Check an actor putting something on (this is the can't put onto something being carried rule):

if the actor encloses the second noun,
stop the action with library message putting it on action number 4 for
the second noun.

Check an actor putting something on (this is the can't put clothes being worn rule):

if the actor is wearing the noun:
issue library message putting it on action number 5 for the noun;
silently try the actor trying taking off the noun;
if the actor is wearing the noun, stop the action;

Check an actor putting something on (this is the can't put if this exceeds carrying capacity rule):

if the second noun provides the property carrying capacity:
if the number of things on the second noun is at least the carrying capacity
of the second noun,
stop the action with library message putting it on action number 6 for
the second noun;

§18. Carry out.

Carry out an actor putting something on (this is the standard putting rule):
now the noun is on the second noun.

§19. Report.

Report an actor putting something on (this is the concise report putting rule):
if the actor is the player and the I6 parser is running multiple actions,
stop the action with library message putting it on action number 7
for the noun;
otherwise continue the action.

Report an actor putting something on (this is the standard report putting rule):
if the actor is the player, issue library message putting it on action
number 8 for the noun;
otherwise issue actor-based library message putting it on action
number 9 for the noun.

§20. Inserting it into.

Inserting it into is an action applying to two things.

The inserting it into action translates into I6 as "Insert".

The specification of the inserting it into action is "By this action, an actor puts something he is holding into a container: for instance, putting a coin into a collection box."

§21. Check.

Check an actor inserting something into (this is the convert insert to drop where possible rule):

if the second noun is down or the actor is in the second noun,
convert to the dropping action on the noun.

Check an actor inserting something into (this is the can't insert what's not held rule):

if the actor is carrying the noun, continue the action;
if the actor is wearing the noun, continue the action;
issue miscellaneous library message number 26 for the noun;
silently try the actor taking the noun;
if the actor is carrying the noun, continue the action;
stop the action.

Check an actor inserting something into (this is the can't insert something into itself rule):

let the noun-CPC be the component parts core of the noun;
let the second-CPC be the component parts core of the second noun;
let the transfer ceiling be the common ancestor of the noun-CPC with the second-CPC;
if the transfer ceiling is the noun-CPC,
stop the action with library message inserting it into action number 5 for
the noun.

Check an actor inserting something into (this is the can't insert into closed containers rule):

if the second noun is a closed container,
stop the action with library message inserting it into action number 3 for
the second noun.

Check an actor inserting something into (this is the can't insert into what's not a container rule):

if the second noun is not a container,
stop the action with library message inserting it into action number 2 for
the second noun.

Check an actor inserting something into (this is the can't insert clothes being worn rule):

if the actor is wearing the noun:
issue library message inserting it into action number 6 for the noun;
silently try the actor trying taking off the noun;
if the actor is wearing the noun, stop the action;

Check an actor inserting something into (this is the can't insert if this exceeds carrying capacity rule):

if the second noun provides the property carrying capacity:
if the number of things in the second noun is at least the carrying capacity
of the second noun,
stop the action with library message inserting it into action number 7 for
the second noun;

§22. Carry out.

Carry out an actor inserting something into (this is the standard inserting rule):
now the noun is in the second noun.

§23. Report.

Report an actor inserting something into (this is the concise report inserting rule):
if the actor is the player and the I6 parser is running multiple actions,
stop the action with library message inserting it into action number 8
for the noun;
otherwise continue the action.

Report an actor inserting something into (this is the standard report inserting rule):
if the actor is the player, issue library message inserting it into action
number 9 for the noun;
otherwise issue actor-based library message inserting it into action number 10 for the noun.

§24. Eating.

Eating is an action applying to one carried thing.

The eating action translates into I6 as "Eat".

The specification of the eating action is "Eating is the only one of the built-in actions which can, in effect, destroy something: the carry out rule removes what's being eaten from play, and nothing in the Standard Rules can then get at it again.

Note that, uncontroversially, one can only eat things with the 'edible' either/or property, and also that, more controversially, one can only eat things currently being held. This means that a player standing next to a bush with berries who types EAT BERRIES will force a '(first taking the berries)' action."

§25. Check.

Check an actor eating (this is the can't eat unless edible rule):

if the noun is not a thing or the noun is not edible,
stop the action with library message eating action number 1 for the noun.

Check an actor eating (this is the can't eat clothing without removing it first rule):

if the actor is wearing the noun:
issue library message dropping action number 3 for the noun;
try the actor trying taking off the noun;
if the actor is wearing the noun, stop the action;

§26. Carry out.

Carry out an actor eating (this is the standard eating rule):

remove the noun from play.

§27. Report.

Report an actor eating (this is the standard report eating rule):

if the actor is the player, issue library message eating action number 2
for the noun;
otherwise issue actor-based library message eating action number 3 for the noun.

§28. Going.

Section SR4/3 - Standard actions which move the actor

Going is an action applying to one visible thing.

The going action translates into I6 as "Go".

The specification of the going action is "This is the action which allows people to move from one room to another, using whatever map connections and doors are to hand. The Standard Rules are written so that the noun can be either a direction or a door in the location of the actor: while the player's commands only lead to going actions with directions as nouns, going actions can also happen as a result of entering actions, and then the noun can indeed be a door."

The going action has a room called the room gone from (matched as "from").

The going action has an object called the room gone to (matched as "to").

The going action has an object called the door gone through (matched as "through").

The going action has an object called the vehicle gone by (matched as "by").

The going action has an object called the thing gone with (matched as "with").

Rule for setting action variables for going (this is the standard set going variables rule):

```

now the thing gone with is the item-pushed-between-rooms;
now the room gone from is the location of the actor;
if the actor is in an enterable vehicle (called the carriage),
    now the vehicle gone by is the carriage;
let the target be nothing;
if the noun is a direction:
    let direction D be the noun;
    let the target be the room-or-door direction D from the room gone from;
otherwise:
    if the noun is a door, let the target be the noun;
if the target is a door:
    now the door gone through is the target;
    now the target is the other side of the target from the room gone from;
now the room gone to is the target.
```

§29. Check.

Check an actor going when the actor is on a supporter (called the chaise)

(this is the stand up before going rule):

issue library message going action number 27 for the chaise;

silently try the actor exiting.

Check an actor going (this is the can't travel in what's not a vehicle rule):

let H be the holder of the actor;

if H is the room gone from, continue the action;

if H is the vehicle gone by, continue the action;

stop the action with library message going action number 1 for H.

Check an actor going (this is the can't go through undescribed doors rule):

if the door gone through is not nothing and the door gone through is undescribed,

stop the action with library message going action number 2 for the room gone from.

Check an actor going (this is the can't go through closed doors rule):

if the door gone through is not nothing and the door gone through is closed:

issue library message going action number 28 for the door gone through;

silently try the actor opening the door gone through;

if the door gone through is open, continue the action;
stop the action.

Check an actor going (this is the determine map connection rule):

let the target be nothing;
if the noun is a direction:
let direction D be the noun;
let the target be the room-or-door direction D from the room gone from;
otherwise:
if the noun is a door, let the target be the noun;
if the target is a door:
now the target is the other side of the target from the room gone from;
now the room gone to is the target.

Check an actor going (this is the can't go that way rule):

if the room gone to is nothing:
if the door gone through is nothing, stop the action with library
message going action number 2 for the room gone from;
stop the action with library message going action number 6 for the door gone through;

§30. Carry out.

Carry out an actor going (this is the move player and vehicle rule):

if the vehicle gone by is nothing,
surreptitiously move the actor to the room gone to during going;
otherwise
surreptitiously move the vehicle gone by to the room gone to during going.

Carry out an actor going (this is the move floating objects rule):

if the actor is the player,
update backdrop positions.

Carry out an actor going (this is the check light in new location rule):

if the actor is the player,
surreptitiously reckon darkness.

§31. Report.

Report an actor going (this is the describe room gone into rule):

if the player is the actor:
produce a room description with going spacing conventions;
otherwise:
if the noun is a direction:
if the location is the room gone from:
if the location is the room gone to:
continue the action;
otherwise:
if the noun is up :
issue actor-based library message going action number 8;
otherwise if the noun is down:
issue actor-based library message going action number 9;
otherwise:
issue actor-based library message going action number 10 for the noun;
otherwise:
let the back way be the opposite of the noun;

```

if the location is the room gone to:
    let the room back the other way be the room back way from the
        location;
    let the room normally this way be the room noun from the
        room gone from;
    if the room back the other way is the room gone from or
        the room back the other way is the room normally this way:
        if the back way is up:
            issue actor-based library message going action number 11;
        otherwise if the back way is down:
            issue actor-based library message going action number 12;
        otherwise:
            issue actor-based library message going action number 13
                for the back way;
    otherwise:
        issue actor-based library message going action number 14;
otherwise:
    if the back way is up :
        issue actor-based library message going action number 15
            for the room gone to;
    otherwise if the back way is down:
        issue actor-based library message going action number 16
            for the room gone to;
    otherwise:
        issue actor-based library message going action number 17
            for the room gone to and the back way;
otherwise if the location is the room gone from:
    issue actor-based library message going action number 18 for the noun;
otherwise:
    issue actor-based library message going action number 19 for the noun;
if the vehicle gone by is not nothing:
    say " ";
    if the vehicle gone by is a supporter, issue actor-based library message
        going action number 20 for the vehicle gone by;
    otherwise issue actor-based library message going action number 21
        for the vehicle gone by;
if the thing gone with is not nothing:
    if the player is within the thing gone with:
        issue actor-based library message going action number 22 for the thing gone with;
    otherwise if the player is within the vehicle gone by:
        issue actor-based library message going action number 23 for the thing gone with;
    otherwise if the location is the room gone from:
        issue actor-based library message going action number 24 for the thing gone with;
    otherwise:
        issue actor-based library message going action number 25 for the thing gone with;
if the player is within the vehicle gone by and the player is not
    within the thing gone with:
    issue actor-based library message going action number 26;
    say ".";
    try looking;
    continue the action;
say ".";

```

§32. Entering.

Entering is an action applying to one thing.

The entering action translates into I6 as "Enter".

The specification of the entering action is "Whereas the going action allows people to move from one location to another in the model world, the entering action is for movement inside a location: for instance, climbing into a cage or sitting on a couch. (Entering is not allowed to change location, so any attempt to enter a door is converted into a going action.) What makes entering trickier than it looks is that the player may try to enter an object which is itself inside, or part of, something else, which might in turn be... and so on. To preserve realism, the implicitly pass through other barriers rule automatically generates entering and exiting actions needed to pass between anything which might be in the way: for instance, in a room with two open cages, an actor in cage A who tries entering cage B first has to perform an exiting action."

Rule for supplying a missing noun while entering (this is the find what to enter rule):

```

    if something enterable (called the box) is in the location,
        now the noun is the box;
    otherwise continue the activity.
  
```

The find what to enter rule is listed last in the for supplying a missing noun rulebook.

§33. Check.

Check an actor entering (this is the convert enter door into go rule):

```

    if the noun is a door, convert to the going action on the noun.
  
```

Check an actor entering (this is the convert enter compass direction into go rule):

```

    if the noun is a direction, convert to the going action on the noun.
  
```

Check an actor entering (this is the can't enter what's already entered rule):

```

    let the local ceiling be the common ancestor of the actor with the noun;
    if the local ceiling is the noun, stop the action with library message
        entering action number 1 for the noun.
  
```

Check an actor entering (this is the can't enter what's not enterable rule):

```

    if the noun is not enterable, stop the action with library message
        entering action number 2 for the noun.
  
```

Check an actor entering (this is the can't enter closed containers rule):

```

    if the noun is a closed container, stop the action with library message
        entering action number 3 for the noun.
  
```

Check an actor entering (this is the can't enter something carried rule):

```

    let the local ceiling be the common ancestor of the actor with the noun;
    if the local ceiling is the actor, stop the action with library message
        entering action number 4 for the noun.
  
```

Check an actor entering (this is the implicitly pass through other barriers rule):

```

    if the holder of the actor is the holder of the noun, continue the action;
    let the local ceiling be the common ancestor of the actor with the noun;
    while the holder of the actor is not the local ceiling:
        let the target be the holder of the actor;
        issue library message entering action number 6 for the target;
        silently try the actor trying exiting;
  
```

```

    if the holder of the actor is the target, stop the action;
if the holder of the actor is the noun, stop the action;
if the holder of the actor is the holder of the noun, continue the action;
let the target be the holder of the noun;
if the noun is part of the target, let the target be the holder of the target;
while the target is a thing:
    if the holder of the target is the local ceiling:
        issue library message entering action number 7 for the target;
        silently try the actor trying entering the target;
        if the holder of the actor is not the target, stop the action;
        convert to the entering action on the noun;
        continue the action;
    let the target be the holder of the target;

```

§34. Carry out.

Carry out an actor entering (this is the standard entering rule):
 surreptitiously move the actor to the noun.

§35. Report.

Report an actor entering (this is the standard report entering rule):
 if the actor is the player:
 issue library message entering action number 5 for the noun;
 otherwise if the noun is a container:
 issue actor-based library message entering action number 8 for the noun;
 otherwise:
 issue actor-based library message entering action number 9 for the noun;
 continue the action.

Report an actor entering (this is the describe contents entered into rule):
 if the actor is the player, describe locale for the noun.

§36. Exiting.

Exiting is an action applying to nothing.

The exiting action translates into I6 as "Exit".

The exiting action has an object called the container exited from (matched as "from").

The specification of the exiting action is "Whereas the going action allows people to move from one location to another in the model world, and the entering action is for movement deeper inside the objects in a location, the exiting action is for movement back out towards the main floor area. Climbing out of a cupboard, for instance, is an exiting action. Exiting when already in the main floor area of a room with a map connection to the outside is converted to a going action. Finally, note that whereas entering works for either containers or supporters, exiting is purely for getting oneself out of containers: if the actor is on top of a supporter instead, an exiting action is converted to the getting off action."

Setting action variables for exiting:

now the container exited from is the holder of the actor.

§37. Check.

Check an actor exiting (this is the convert exit into go out rule):

let the local room be the location of the actor;

if the container exited from is the local room:

if the room-or-door outside from the local room is not nothing,
convert to the going action on the outside;

Check an actor exiting (this is the can't exit when not inside anything rule):

let the local room be the location of the actor;

if the container exited from is the local room, stop the action with
library message exiting action number 1 for the actor.

Check an actor exiting (this is the can't exit closed containers rule):

if the actor is in a closed container (called the cage), stop the action
with library message exiting action number 2 for the cage.

Check an actor exiting (this is the convert exit into get off rule):

if the actor is on a supporter (called the platform),
convert to the getting off action on the platform.

§38. Carry out.

Carry out an actor exiting (this is the standard exiting rule):

let the former exterior be the not-counting-parts holder of the container exited from;
surreptitiously move the actor to the former exterior.

§39. Report.

Report an actor exiting (this is the standard report exiting rule):

if the actor is the player:

 issue library message exiting action number 3 for the container exited from;

otherwise:

 issue actor-based library message exiting action number 6 for the container exited from;

continue the action.

Report an actor exiting (this is the describe room emerged into rule):

if the actor is the player,

 produce a room description with going spacing conventions.

§40. Getting off.

Getting off is an action applying to one thing.

The getting off action translates into I6 as "GetOff".

The specification of the getting off action is "The getting off action is for actors who are currently on top of a supporter: perhaps standing on a platform, but maybe only sitting on a chair or even lying down in bed. Unlike the similar exiting action, getting off takes a noun: the platform, chair, bed or what have you."

§41. Check.

Check an actor getting off (this is the can't get off things rule):

- if the actor is on the noun, continue the action;
- if the actor is carried by the noun, continue the action;
- stop the action with library message getting off action number 1 for the noun.

§42. Carry out.

Carry out an actor getting off (this is the standard getting off rule):

- let the former exterior be the not-counting-parts holder of the noun;
- surreptitiously move the actor to the former exterior.

§43. Report.

Report an actor getting off (this is the standard report getting off rule):

- if the actor is the player:
 - issue library message exiting action number 3 for the noun;
- otherwise:
 - issue actor-based library message exiting action number 5 for the noun;
- continue the action.

Report an actor getting off (this is the describe room stood up into rule):

- if the actor is the player,
 - produce a room description with going spacing conventions.

§44. Looking.

Section SR4/4 - Standard actions concerning the actor's vision

Looking is an action applying to nothing.

The looking action translates into I6 as "Look".

The specification of the looking action is "The looking action describes the player's current room and any visible items, but is made more complicated by the problem of visibility. Inform calculates this by dividing the room into visibility levels. For an actor on the floor of a room, there is only one such level: the room itself. But an actor sitting on a chair inside a packing case which is itself on a gantry would have four visibility levels: chair, case, gantry, room. The looking rules use a special phrase, 'the visibility-holder of X', to go up from one level to the next: thus the visibility-holder of the case is the gantry.

The 'visibility level count' is the number of levels which the player can actually see, and the 'visibility ceiling' is the uppermost visible level. For a player standing on the floor of a lighted room, this will be a count of 1 with the ceiling set to the room. But a player sitting on a chair in a closed opaque packing case would have visibility level count 2, and visibility ceiling equal to the case. Moreover, light has to be available in order to see anything at all: if the player is in darkness, the level count is 0 and the ceiling is nothing.

Finally, note that several actions other than looking also produce room descriptions in some cases. The most familiar is going, but exiting a container or getting off a supporter will also generate a room description. (The phrase used by the relevant rules is 'produce a room description with going spacing conventions' and carry out or report rules for newly written actions are welcome to use this too if they would like to. The spacing conventions affect paragraph division, and note that the main description paragraph may be omitted for a place not newly visited, depending on the VERBOSE settings.) Room descriptions like this are produced by running the check, carry out and report rules for looking, but are not subject to before, instead or after rules: so they do not count as a new action. The looking variable 'room-describing action' holds the action name of the reason a room description is currently being made: if the player typed LOOK, this will indeed be set to the looking action, but if we're describing a room just reached by GO EAST, say, it will be set to the going action. This can be used to customise carry out looking rules so that different forms of description are used on going to a room as compared with looking around while already there."

The looking action has an action name called the room-describing action.

The looking action has a truth state called abbreviated form allowed.

The looking action has a number called the visibility level count.

The looking action has an object called the visibility ceiling.

Setting action variables for looking (this is the determine visibility ceiling rule):

```
if the actor is the player, calculate visibility ceiling at low level;
now the visibility level count is the visibility ceiling count calculated;
now the visibility ceiling is the visibility ceiling calculated;
now the room-describing action is the looking action.
```

§45. Carry out.

Carry out looking (this is the room description heading rule):

```

say bold type;
if the visibility level count is 0:
    begin the printing the name of a dark room activity;
    if handling the printing the name of a dark room activity,
        issue miscellaneous library message number 71;
    end the printing the name of a dark room activity;
otherwise if the visibility ceiling is the location:
    say "[visibility ceiling]";
otherwise:
    say "[The visibility ceiling]";
say roman type;
let intermediate level be the visibility-holder of the actor;
repeat with intermediate level count running from 2 to the visibility level count:
    issue library message looking action number 8 for the intermediate level;
    let the intermediate level be the visibility-holder of the intermediate level;
say line break;
say run paragraph on with special look spacing.

```

Carry out looking (this is the room description body text rule):

```

if the visibility level count is 0:
    if set to abbreviated room descriptions, continue the action;
    if set to sometimes abbreviated room descriptions and
        abbreviated form allowed is true and
        darkness witnessed is true,
        continue the action;
    begin the printing the description of a dark room activity;
    if handling the printing the description of a dark room activity,
        issue miscellaneous library message number 17;
    end the printing the description of a dark room activity;
otherwise if the visibility ceiling is the location:
    if set to abbreviated room descriptions, continue the action;
    if set to sometimes abbreviated room descriptions and abbreviated form
        allowed is true and the location is visited, continue the action;
    print the location's description;

```

Carry out looking (this is the room description paragraphs about objects rule):

```

if the visibility level count is greater than 0:
    let the intermediate position be the actor;
    let the IP count be the visibility level count;
    while the IP count is greater than 0:
        now the intermediate position is marked for listing;
        let the intermediate position be the visibility-holder of the
            intermediate position;
        decrease the IP count by 1;
    let the top-down IP count be the visibility level count;
    while the top-down IP count is greater than 0:
        let the intermediate position be the actor;
        let the IP count be 0;
        while the IP count is less than the top-down IP count:
            let the intermediate position be the visibility-holder of the
                intermediate position;
            increase the IP count by 1;

```

```
    [if we ever support I6-style inside descriptions, here's where]
    describe locale for the intermediate position;
    decrease the top-down IP count by 1;
continue the action;
Carry out looking (this is the check new arrival rule):
  if in darkness:
    now the darkness witnessed is true;
  otherwise:
    if the location is a room, now the location is visited;
```

§46. Report.

```
Report an actor looking (this is the other people looking rule):
  if the actor is not the player,
    issue actor-based library message looking action number 9.
```

§47. Examining.

Examining is an action applying to one visible thing and requiring light. The examining action translates into I6 as "Examine".

The specification of the examining action is "The act of looking closely at something. Note that the noun could be either a direction or a thing, which is why the Standard Rules include the 'examine directions rule' to deal with directions: it simply says 'You see nothing unexpected in that direction.' and stops the action. (If you would like to handle directions differently, list another rule instead of this one in the carry out examining rules.)

Some things have no description property, or rather, have only a blank text as one. It's possible that something interesting may be said anyway (see the rules for directions, containers, supporters and devices), but if not, we give up with a bland response. This is done by the examine undescribed things rule."

The examining action has a truth state called examine text printed.

§48. Carry out.

Carry out examining (this is the standard examining rule):

```
if the noun provides the property description and the description of the noun is not "":
    say "[the description of the noun][line break]";
    now examine text printed is true.
```

Carry out examining (this is the examine directions rule):

```
if the noun is a direction:
    issue library message examining action number 5 for the noun;
    now examine text printed is true.
```

Carry out examining (this is the examine containers rule):

```
if the noun is a container:
    if the noun is open or the noun is transparent:
        if something described which is not scenery is in the noun and something which
        is not the player is in the noun:
            issue library message searching action number 7 for the noun;
            now examine text printed is true;
        otherwise if examine text printed is false:
            if the player is in the noun:
                make no decision;
            issue library message searching action number 6 for the noun;
            now examine text printed is true;
```

Carry out examining (this is the examine supporters rule):

```
if the noun is a supporter:
    if something described which is not scenery is on the noun and something which is
    not the player is on the noun:
        issue library message looking action number 4 for the noun;
        now examine text printed is true.
```

Carry out examining (this is the examine devices rule):

```
if the noun is a device:
    issue library message examining action number 3 for the noun;
    now examine text printed is true.
```

Carry out examining (this is the examine undescribed things rule):

```
if examine text printed is false:
    issue library message examining action number 2 for the noun.
```

§49. Report.

Report an actor examining (this is the report other people examining rule):
if the actor is not the player,
 issue actor-based library message examining action number 4 for the noun.

§50. Looking under.

Looking under is an action applying to one visible thing and requiring light. The looking under action translates into I6 as "LookUnder".

The specification of the looking under action is "The standard Inform world model does not have a concept of things being under other things, so this action is only minimally provided by the Standard Rules, but it exists here for traditional reasons (and because, after all, LOOK UNDER TABLE is the sort of command which ought to be recognised even if it does nothing useful). The action ordinarily either tells the player he finds nothing of interest, or reports that somebody else has looked under something.

The usual way to make this action do something useful is to write a rule like 'Instead of looking under the cabinet for the first time: now the player has the silver key; say ...' and so on."

§51. Carry out.

Carry out an actor looking under (this is the standard looking under rule):
 stop the action with library message looking under action number 2 for
 the noun.

§52. Report.

Report an actor looking under (this is the report other people looking under rule):
 if the actor is not the player,
 issue actor-based library message looking under action number 3 for the noun.

§53. Searching.

Searching is an action applying to one thing and requiring light.
The searching action translates into I6 as "Search".

The specification of the searching action is "Searching looks at the contents of an open or transparent container, or at the items on top of a supporter. These are often mentioned in room descriptions already, and then the action is unnecessary, but that wouldn't be true for something like a kitchen cupboard which is scenery - mentioned in passing in a room description, but not made a fuss of. Searching such a cupboard would then, by listing its contents, give the player more information than the ordinary room description shows.

The usual check rules restrict searching to containers and supporters: so the Standard Rules do not allow the searching of people, for instance. But it is easy to add instead rules ('Instead of searching Dr Watson: ...') or even a new carry out rule ('Check searching someone (called the suspect): ...') to extend the way searching normally works."

§54. Check.

Check an actor searching (this is the can't search unless container or supporter rule):
if the noun is not a container and the noun is not a supporter,
stop the action with library message searching action number 4 for
the noun.

Check an actor searching (this is the can't search closed opaque containers rule):
if the noun is a closed opaque container,
stop the action with library message searching action number 5 for
the noun.

§55. Report.

Report searching a container (this is the standard search containers rule):
if the noun contains a described thing which is not scenery,
issue library message searching action number 7 for the noun;
otherwise
issue library message searching action number 6 for the noun.

Report searching a supporter (this is the standard search supporters rule):
if the noun supports a described thing which is not scenery,
issue library message searching action number 3 for the noun;
otherwise
issue library message searching action number 2 for the noun.

Report an actor searching (this is the report other people searching rule):
if the actor is not the player,
issue actor-based library message searching action number 8 for the noun.

§56. Consulting it about.

Consulting it about is an action applying to one thing and one topic.

The consulting it about action translates into I6 as "Consult".

The specification of the consulting it about action is "Consulting is a very flexible and potentially powerful action, but only because it leaves almost all of the work to the author to deal with directly. The idea is for it to respond to commands such as LOOK UP HENRY FITZROY IN HISTORY BOOK, where the topic would be the snippet of command HENRY FITZROY and the thing would be the book.

The Standard Rules simply parry such requests by saying that the player finds nothing of interest. All interesting responses must be provided by the author, using rules like 'Instead of consulting the history book about...'

§57. Report.

Report an actor consulting something about (this is the block consulting rule):

if the actor is the player,

 issue library message consulting it about action number 1 for the noun;

otherwise

 issue actor-based library message consulting it about action number 2 for the noun.

§58. Locking it with.

Section SR4/5 - Standard actions which change the state of things

Locking it with is an action applying to one thing and one carried thing.
The locking it with action translates into I6 as "Lock".

The specification of the locking it with action is "Locking is the act of using an object such as a key to ensure that something such as a door or container cannot be opened unless first unlocked. (Only closed things can be locked.)"

Locking can be performed on any kind of thing which provides the either/or properties lockable, locked, openable and open. The 'can't lock without a lock rule' tests to see if the noun both provides the lockable property, and if it is in fact lockable: it is then assumed that the other properties can safely be checked. In the Standard Rules, the container and door kinds both satisfy these requirements.

We can create a new kind on which opening, closing, locking and unlocking will work thus: 'A briefcase is a kind of thing. A briefcase can be openable. A briefcase can be open. A briefcase can be lockable. A briefcase can be locked. A briefcase is usually openable, lockable, open and unlocked.'

Inform checks whether the key fits using the 'can't lock without the correct key rule'. To satisfy this, the actor must be directly holding the second noun, and it must be the current value of the 'matching key' property for the noun. (This property is seldom referred to directly because it is automatically set by assertions like 'The silver key unlocks the wicket gate.')

The Standard Rules provide locking and unlocking actions at a fairly basic level: they can be much enhanced using the extension Locksmith by Emily Short, which is included with all distributions of Inform."

§59. Check.

Check an actor locking something with (this is the can't lock without a lock rule):

- if the noun provides the property lockable and the noun is lockable,
 - continue the action;
 - stop the action with library message locking it with action number 1 for the noun.

Check an actor locking something with (this is the can't lock what's already locked rule):

- if the noun is locked,
 - stop the action with library message locking it with action number 2 for the noun.

Check an actor locking something with (this is the can't lock what's open rule):

- if the noun is open,
 - stop the action with library message locking it with action number 3 for the noun.

Check an actor locking something with (this is the can't lock without the correct key rule):

- if the holder of the second noun is not the actor or
 - the noun does not provide the property matching key or
 - the matching key of the noun is not the second noun,
 - stop the action with library message locking it with action number 4 for the second noun.

§60. Carry out.

Carry out an actor locking something with (this is the standard locking rule):
now the noun is locked.

§61. Report.

Report an actor locking something with (this is the standard report locking rule):
if the actor is the player:
 issue library message locking it with action number 5 for the noun;
otherwise:
 if the actor is visible, issue actor-based library message locking it with
 action number 6 for the noun;

§62. Unlocking it with.

Unlocking it with is an action applying to one thing and one carried thing. The unlocking it with action translates into I6 as "Unlock".

The specification of the unlocking it with action is "Unlocking undoes the effect of locking, and renders the noun openable again provided that the actor is carrying the right key (which must be the second noun).

Unlocking can be performed on any kind of thing which provides the either/or properties lockable, locked, openable and open. The 'can't unlock without a lock rule' tests to see if the noun both provides the lockable property, and if it is in fact lockable: it is then assumed that the other properties can safely be checked. In the Standard Rules, the container and door kinds both satisfy these requirements.

We can create a new kind on which opening, closing, locking and unlocking will work thus: 'A briefcase is a kind of thing. A briefcase can be openable. A briefcase can be open. A briefcase can be lockable. A briefcase can be locked. A briefcase is usually openable, lockable, open and unlocked.'

Inform checks whether the key fits using the 'can't unlock without the correct key rule'. To satisfy this, the actor must be directly holding the second noun, and it must be the current value of the 'matching key' property for the noun. (This property is seldom referred to directly because it is automatically set by assertions like 'The silver key unlocks the wicket gate.')

The Standard Rules provide locking and unlocking actions at a fairly basic level: they can be much enhanced using the extension Locksmith by Emily Short, which is included with all distributions of Inform."

§63. Check.

Check an actor unlocking something with (this is the can't unlock without a lock rule):

if the noun provides the property lockable and the noun is lockable,
continue the action;

stop the action with library message unlocking it with action number 1 for the noun.

Check an actor unlocking something with (this is the can't unlock what's already unlocked rule):

if the noun is not locked,

stop the action with library message unlocking it with action number 2 for the noun.

Check an actor unlocking something with (this is the can't unlock without the correct key rule):

if the holder of the second noun is not the actor or

the noun does not provide the property matching key or

the matching key of the noun is not the second noun,

stop the action with library message unlocking it with action number 3 for the
second noun.

§64. Carry out.

Carry out an actor unlocking something with (this is the standard unlocking rule):

now the noun is not locked.

§65. Report.

Report an actor unlocking something with (this is the standard report unlocking rule):

if the actor is the player:

 issue library message unlocking it with action number 4 for the noun;

otherwise:

 if the actor is visible, issue actor-based library message unlocking it with
 action number 5 for the noun;

§66. Switching on.

Switching on is an action applying to one thing.

The switching on action translates into I6 as "SwitchOn".

The specification of the switching on action is "The switching on and switching off actions are for the simplest kind of machinery operation: they are for objects representing machines (or more likely parts of machines), which are considered to be either off or on at any given time.

The actions are intended to be used where the noun is a device, but in fact they could work just as well with any kind which can be 'switched on' or 'switched off'."

§67. Check.

Check an actor switching on (this is the can't switch on unless switchable rule):

if the noun provides the property switched on, continue the action;

stop the action with library message switching on action number 1 for the noun.

Check an actor switching on (this is the can't switch on what's already on rule):

if the noun is switched on,

stop the action with library message switching on action number 2 for the noun.

§68. Carry out.

Carry out an actor switching on (this is the standard switching on rule):

now the noun is switched on.

§69. Report.

Report an actor switching on (this is the standard report switching on rule):

if the actor is the player, issue library message switching on action number 3
for the noun;

otherwise issue actor-based library message switching on action number 4 for the noun;

§70. Switching off.

Switching off is an action applying to one thing.

The switching off action translates into I6 as "SwitchOff".

The specification of the switching off action is "The switching off and switching on actions are for the simplest kind of machinery operation: they are for objects representing machines (or more likely parts of machines), which are considered to be either off or on at any given time.

The actions are intended to be used where the noun is a device, but in fact they could work just as well with any kind which can be 'switched on' or 'switched off'."

§71. Check.

Check an actor switching off (this is the can't switch off unless switchable rule):

if the noun provides the property switched on, continue the action;

stop the action with library message switching off action number 1 for the noun.

Check an actor switching off (this is the can't switch off what's already off rule):

if the noun is switched off,

stop the action with library message switching off action number 2 for the noun.

§72. Carry out.

Carry out an actor switching off (this is the standard switching off rule):

now the noun is switched off.

§73. Report.

Report an actor switching off (this is the standard report switching off rule):

if the actor is the player, issue library message switching off action number 3
for the noun;

otherwise issue actor-based library message switching off action number 4 for the noun;

§74. Opening.

Opening is an action applying to one thing.

The opening action translates into I6 as "Open".

The specification of the opening action is "Opening makes something no longer a physical barrier. The action can be performed on any kind of thing which provides the either/or properties openable and open. The 'can't open unless openable rule' tests to see if the noun both can be and actually is openable. (It is assumed that anything which can be openable can also be open.) In the Standard Rules, the container and door kinds both satisfy these requirements.

In the event that the thing to be opened is also lockable, we are forbidden to open it when it is locked. Both containers and doors can be lockable, but the opening and closing actions would also work fine with kinds which cannot be.

We can create a new kind on which opening and closing will work thus:

'A case file is a kind of thing. A case file can be openable.

A case file can be open. A case file is usually openable and closed.'

The meaning of open and closed is different for different kinds of thing. When a container is closed, that means people outside cannot reach in, and vice versa; when a door is closed, people cannot use the 'going' action to pass through it. If we were to create a new kind such as 'case file', we would also need to write rules to make the open and closed properties interesting for this kind."

§75. Check.

Check an actor opening (this is the can't open unless openable rule):

if the noun provides the property openable and the noun is openable,
continue the action;

stop the action with library message opening action number 1 for the noun.

Check an actor opening (this is the can't open what's locked rule):

if the noun provides the property lockable and the noun is locked,
stop the action with library message opening action number 2 for the noun.

Check an actor opening (this is the can't open what's already open rule):

if the noun is open,
stop the action with library message opening action number 3 for the noun.

§76. Carry out.

Carry out an actor opening (this is the standard opening rule):

now the noun is open.

§77. Report.

Report an actor opening (this is the reveal any newly visible interior rule):

```
if the actor is the player and
    the noun is an opaque container and
    the first thing held by the noun is not nothing and
    the noun does not enclose the actor,
    stop the action with library message opening action number 4 for the noun.
```

Report an actor opening (this is the standard report opening rule):

```
if the actor is the player:
    issue library message opening action number 5 for the noun;
otherwise if the player can see the actor:
    issue actor-based library message opening action number 6 for the noun;
otherwise:
    issue actor-based library message opening action number 7 for the noun;
```

§78. Closing.

Closing is an action applying to one thing.

The closing action translates into I6 as "Close".

The specification of the closing action is "Closing makes something into a physical barrier. The action can be performed on any kind of thing which provides the either/or properties openable and open. The 'can't close unless openable rule' tests to see if the noun both can be and actually is openable. (It is assumed that anything which can be openable can also be open, and hence can also be closed.) In the Standard Rules, the container and door kinds both satisfy these requirements.

We can create a new kind on which opening and closing will work thus:

'A case file is a kind of thing. A case file can be openable.

A case file can be open. A case file is usually openable and closed.'

The meaning of open and closed is different for different kinds of thing.

When a container is closed, that means people outside cannot reach in, and vice versa; when a door is closed, people cannot use the 'going' action to pass through it. If we were to create a new kind such as 'case file', we would also need to write rules to make the open and closed properties interesting for this kind."

§79. Check.

Check an actor closing (this is the can't close unless openable rule):

if the noun provides the property openable and the noun is openable,
continue the action;

stop the action with library message closing action number 1 for the noun.

Check an actor closing (this is the can't close what's already closed rule):

if the noun is closed,

stop the action with library message closing action number 2 for the noun.

§80. Carry out.

Carry out an actor closing (this is the standard closing rule):

now the noun is closed.

§81. Report.

Report an actor closing (this is the standard report closing rule):

if the actor is the player:

issue library message closing action number 3 for the noun;

otherwise if the player can see the actor:

issue actor-based library message closing action number 4 for the noun;

otherwise:

issue actor-based library message closing action number 5 for the noun;

§82. Wearing.

Wearing is an action applying to one carried thing.
The wearing action translates into I6 as "Wear".

The specification of the wearing action is "The Standard Rules give Inform only a simple model of clothing. A thing can be worn only if it has the either/or property of being 'wearable'. (Typing a sentence like 'Mr Jones wears the Homburg hat.' automatically implies that the hat is wearable, which is why we only seldom need to use the word 'wearable' directly.) There is no checking of how much or how little any actor is wearing, or how incongruous this may appear: nor any distinction between under or over-clothes.

To put on an article of clothing, the actor must be directly carrying it, as enforced by the 'can't wear what's not held rule'."

§83. Check.

Check an actor wearing (this is the can't wear what's not clothing rule):

if the noun is not a thing or the noun is not wearable,
stop the action with library message wearing action number 1 for the noun.

Check an actor wearing (this is the can't wear what's not held rule):

if the holder of the noun is not the actor,
stop the action with library message wearing action number 2 for the noun.

Check an actor wearing (this is the can't wear what's already worn rule):

if the actor is wearing the noun,
stop the action with library message wearing action number 3 for the noun.

§84. Carry out.

Carry out an actor wearing (this is the standard wearing rule):

now the actor wears the noun.

§85. Report.

Report an actor wearing (this is the standard report wearing rule):

if the actor is the player, issue library message wearing action number 4
for the noun;
otherwise issue actor-based library message wearing action number 5
for the noun.

§86. Taking off.

Taking off is an action applying to one carried thing.
The taking off action translates into I6 as "Disrobe".

The specification of the taking off action is "The Standard Rules give Inform only a simple model of clothing. A thing can be worn only if it has the either/or property of being 'wearable'. (Typing a sentence like 'Mr Jones wears the Homburg hat.' automatically implies that the hat is wearable, which is why we only seldom need to use the word 'wearable' directly.) There is no checking of how much or how little any actor is wearing, or how incongruous this may appear: nor any distinction between under or over-clothes.

When an article of clothing is taken off, it becomes a thing directly carried by its former wearer, rather than being (say) dropped onto the floor."

§87. Check.

Check an actor taking off (this is the can't take off what's not worn rule):
if the actor is not wearing the noun,
stop the action with library message taking off action number 1 for the noun.

§88. Carry out.

Carry out an actor taking off (this is the standard taking off rule):
now the actor carries the noun.

§89. Report.

Report an actor taking off (this is the standard report taking off rule):
if the actor is the player, issue library message taking off action number 2
for the noun;
otherwise issue actor-based library message taking off action number 3 for the noun.

§90. Giving it to.

Section SR4/6 - Standard actions concerning other people

Giving it to is an action applying to one carried thing and one thing.

The giving it to action translates into I6 as "Give".

The specification of the giving it to action is "This action is indexed by Inform under 'Actions concerning other people', but it could just as easily have gone under 'Actions concerning the actor's possessions' because - like dropping, putting it on or inserting it into - this is an action which gets rid of something being carried.

The Standard Rules implement this action fully - if it reaches the carry out and report rulebooks, then the item is indeed transferred to the recipient, and this is properly reported. But giving something to somebody is not like putting something on a shelf: the recipient has to agree. The final check rule, the 'block giving rule', assumes that the recipient does not consent - so the gift fails to happen. The way to make the giving action use its abilities fully is to replace the block giving rule with a rule which makes a more sophisticated decision about who will accept what from whom, and only blocks some attempts, letting others run on into the carry out and report rules."

§91. Check.

Check an actor giving something to (this is the can't give what you haven't got rule):
 if the actor is not the holder of the noun,
 stop the action with library message giving it to action number 1 for the noun.

Check an actor giving something to (this is the can't give to yourself rule):
 if the actor is the second noun,
 stop the action with library message giving it to action number 2 for the noun.

Check an actor giving something to (this is the can't give to a non-person rule):
 if the second noun is not a person,
 stop the action with library message giving it to action number 4 for the second noun.

Check an actor giving something to (this is the can't give clothes being worn rule):
 if the actor is wearing the noun:
 issue library message dropping action number 3 for the noun;
 silently try the actor trying taking off the noun;
 if the actor is wearing the noun, stop the action;

Check an actor giving something to (this is the block giving rule):
 stop the action with library message giving it to action number 3 for the second noun.

§92. Carry out.

Carry out an actor giving something to (this is the standard giving rule):
 move the noun to the second noun.

§93. Report.

Report an actor giving something to (this is the standard report giving rule):

if the actor is the player:

 issue library message giving it to action number 5 for the noun;

otherwise if the second noun is the player:

 issue actor-based library message giving it to action number 6 for the noun;

otherwise:

 issue actor-based library message giving it to action number 7 for the noun;

§94. Showing it to.

Showing it to is an action applying to one carried thing and one visible thing. The showing it to action translates into I6 as "Show".

The specification of the showing it to action is "Anyone can show anyone else something which they are carrying, but not some nearby piece of scenery, say - so this action is suitable for showing the emerald locket to Katarina, but not showing the Orange River Rock Room to Mr Douglas.

The Standard Rules implement this action in only a minimal way, checking that it makes sense but then blocking all such attempts with a message such as 'Katarina is not interested.' - this is the task of the 'block showing rule'. As a result, there are no carry out or report rules. To make it into a systematic and interesting action, we would need to unlist the block showing rule and then to write carry out and report rules: but usually for IF purposes we only need to make a handful of special cases of showing work properly, and for those we can simply write Instead rules to handle them."

§95. Check.

Check an actor showing something to (this is the can't show what you haven't got rule):

```
if the actor is not the holder of the noun,
    stop the action with library message showing it to action number 1
    for the noun.
```

Check an actor showing something to (this is the convert show to yourself to examine rule):

```
if the actor is the second noun,
    convert to the examining action on the noun.
```

Check an actor showing something to (this is the block showing rule):

```
stop the action with library message showing it to action number 2
for the second noun.
```

§96. Waking.

Waking is an action applying to one thing.

The waking action translates into I6 as "WakeOther".

The specification of the waking action is "This is the act of jostling a sleeping person to wake him or her up, and it finds its way into the Standard Rules only for historical reasons. Inform does not by default provide any model for people being asleep or awake, so this action does not do anything in the standard implementation: instead, it is always stopped by the block waking rule."

§97. Check.

Check an actor waking (this is the block waking rule):

stop the action with library message waking action number 1 for the noun.

§98. Throwing it at.

Throwing it at is an action applying to one carried thing and one visible thing. The throwing it at action translates into I6 as "ThrowAt".

The specification of the throwing it at action is "Throwing something at someone or something is difficult for Inform to model. So many considerations apply: just because the actor can see the target, does it follow that the target can accurately hit it? What if the projectile is heavy, like an anvil, or something not easily aimable, like a feather? What if there is a barrier in the way, like a cage with bars spaced so that only items of a certain size get through? And then: what should happen as a result? Will the projectile break, or do damage, or fall to the floor, or into a container or onto a supporter? And so on.

Because it seems hopeless to try to model this in any general way, Inform instead provides the action for the user to attach specific rules to. The check rules in the Standard Rules simply require that the projectile is not an item of clothing still worn (this will be relevant for women attending a Tom Jones concert) but then, in either the 'futile to throw things at inanimate objects rule' or the 'block throwing at rule', will refuse to carry out the action with a bland message.

To make throwing do something, then, we must either write Instead rules for special circumstances, or else unlist these check rules and write suitable carry out and report rules to pick up the thread."

§99. Check.

Check an actor throwing something at (this is the implicitly remove thrown clothing rule):

```
if the actor is wearing the noun:
    issue library message dropping action number 3 for the noun;
    silently try the actor trying taking off the noun;
    if the actor is wearing the noun, stop the action;
```

Check an actor throwing something at (this is the futile to throw things at inanimate objects rule):

```
if the second noun is not a person,
    stop the action with library message throwing it at action number 1
    for the second noun.
```

Check an actor throwing something at (this is the block throwing at rule):

```
stop the action with library message throwing it at action number 2
for the noun.
```

§100. Attacking.

Attacking is an action applying to one thing.

The attacking action translates into I6 as "Attack".

The specification of the attacking action is "Violence is seldom the answer, and attempts to attack another person are normally blocked as being unrealistic or not seriously meant. (I might find a shop assistant annoying, but IF is not Grand Theft Auto, and responding by killing him is not really one of my options.) So the Standard Rules simply block attempts to fight people, but the action exists for rules to make exceptions."

§101. Check.

Check an actor attacking (this is the block attacking rule):

stop the action with library message attacking action number 1 for the noun.

§102. Kissing.

Kissing is an action applying to one thing.

The kissing action translates into I6 as "Kiss".

The specification of the kissing action is "Possibly because Inform was originally written by an Englishman, attempts at kissing another person are normally blocked as being unrealistic or not seriously meant. So the Standard Rules simply block attempts to kiss people, but the action exists for rules to make exceptions."

§103. Check.

Check an actor kissing (this is the kissing yourself rule):

if the noun is the actor,

stop the action with library message touching action number 3 for the noun.

Check an actor kissing (this is the block kissing rule):

stop the action with library message kissing action number 1 for the noun.

§104. Answering it that.

Answering it that is an action applying to one thing and one topic.
The answering it that action translates into I6 as "Answer".

The specification of the answering it that action is "The Standard Rules do not include any systematic way to handle conversation: instead, Inform is set up so that it is as easy as we can make it to write specific rules handling speech in particular games, and so that if no such rules are written then all attempts to communicate are gracefully if not very interestingly rejected.

The topic here can be any double-quoted text, which can itself contain tokens in square brackets: see the documentation on Understanding.

Answering is an action existing so that the player can say something free-form to somebody else. A convention of IF is that a command such as DAPHNE, TAKE MASK is a request to Daphne to perform an action: if the persuasion rules in force mean that she consents, the action 'Daphne taking the mask' does indeed then result. But if the player types DAPHNE, 12375 or DAPHNE, GREAT HEAVENS - or anything else not making sense as a command - the action 'answering Daphne that ...' will be generated.

The name of the action arises because it is also caused by typing, say, ANSWER 12375 when Daphne (say) has asked a question."

§105. Report.

Report an actor answering something that (this is the block answering rule):
stop the action with library message answering it that action number 1
for the noun.

§106. Telling it about.

Telling it about is an action applying to one thing and one topic.
The telling it about action translates into I6 as "Tell".

The specification of the telling it about action is "The Standard Rules do not include any systematic way to handle conversation: instead, Inform is set up so that it is as easy as we can make it to write specific rules handling speech in particular games, and so that if no such rules are written then all attempts to communicate are gracefully if not very interestingly rejected.

The topic here can be any double-quoted text, which can itself contain tokens in square brackets: see the documentation on Understanding.

Telling is an action existing only to catch commands like TELL ALEX ABOUT GUITAR. Customarily in IF, such a command is shorthand which the player accepts as a conventional form: it means 'tell Alex what I now know about the guitar' and would make sense if the player had himself recently discovered something significant about the guitar which might interest Alex."

§107. Check.

Check an actor telling something about (this is the telling yourself rule):
 if the actor is the noun,
 stop the action with library message telling it about action number 1
 for the noun.

§108. Report.

Report an actor telling something about (this is the block telling rule):
 stop the action with library message telling it about action number 2
 for the noun.

§109. Asking it about.

Asking it about is an action applying to one thing and one topic.
The asking it about action translates into I6 as "Ask".

The specification of the asking it about action is "The Standard Rules do not include any systematic way to handle conversation: instead, Inform is set up so that it is as easy as we can make it to write specific rules handling speech in particular games, and so that if no such rules are written then all attempts to communicate are gracefully if not very interestingly rejected.

The topic here can be any double-quoted text, which can itself contain tokens in square brackets: see the documentation on Understanding.

Asking is an action existing only to catch commands like ASK STEPHEN ABOUT PENELOPE. Customarily in IF, such a command is shorthand which the player accepts as a conventional form: it means 'engage Mary in conversation and try to find out what she might know about'. It's understood as a convention of the genre that Mary should not be expected to respond in cases where there is no reason to suppose that she has anything relevant to pass on - ASK JANE ABOUT RICE PUDDING, for instance, need not conjure up a recipe even if Jane is a 19th-century servant and therefore almost certainly knows one."

§110. Report.

Report an actor asking something about (this is the block asking rule):
stop the action with library message asking it about action number 1
for the noun.

§111. Asking it for.

Asking it for is an action applying to two things.

The asking it for action translates into I6 as "AskFor".

The specification of the asking it for action is "The Standard Rules do not include any systematic way to handle conversation, but this is action is not quite conversation: it doesn't involve any spoken text as such. It exists to catch commands like ASK SALLY FOR THE EGG WHISK, where the whisk is something which Sally has and the player can see.

Slightly oddly, but for historical reasons, an actor asking himself for something is treated to an inventory listing instead. All other cases are converted to the giving action: that is, ASK SALLY FOR THE EGG WHISK is treated as if it were SALLY, GIVE ME THE EGG WHISK - an action for Sally to perform and which then follows rules for giving.

To ask for information or something intangible, see the asking it about action."

§112. Check.

Check an actor asking something for (this is the asking yourself for something rule):
 if the actor is the noun and the actor is the player,
 try taking inventory instead.

Check an actor asking something for (this is the translate asking for to giving rule):
 convert to request of the noun to perform giving it to action with the
 second noun and the actor.

§113. Waiting.

Section SR4/7 - Standard actions which are checked but then do nothing unless rules intervene

Waiting is an action applying to nothing.

The waiting action translates into I6 as "Wait".

The specification of the waiting action is "The inaction action: where would we be without waiting? Waiting does not cause time to pass by - that happens anyway - but represents a positive choice by the actor not to fill that time. It is an action so that rules can be attached to it: for instance, we could imagine that a player who consciously decides to sit and wait might notice something which a busy player does not, and we could write a rule accordingly.

Note the absence of check or carry out rules - anyone can wait, at any time, and it makes nothing happen."

§114. Report.

Report an actor waiting (this is the standard report waiting rule):

```
if the actor is the player, stop the action with library message waiting
    action number 1 for the actor;
issue actor-based library message waiting action number 2.
```

§115. Touching.

Touching is an action applying to one thing.

The touching action translates into I6 as "Touch".

The specification of the touching action is "Touching is just that, touching something without applying pressure: a touch-sensitive screen or a living creature might react, but a standard push-button or lever will probably not.

In the Standard Rules there are no check touching rules, since touchability is already a requirement of the noun for the action anyway, and no carry out rules because nothing in the standard Inform world model reacts to a mere touch - though report rules do mean that attempts to touch other people provoke a special reply."

§116. Report.

Report an actor touching (this is the report touching yourself rule):

```

if the noun is the actor:
    if the actor is the player, issue library message touching action number 3
        for the noun;
    otherwise issue actor-based library message touching action number 4;
    stop the action;
continue the action.

```

Report an actor touching (this is the report touching other people rule):

```

if the noun is a person:
    if the actor is the player:
        issue library message touching action number 1 for the noun;
    otherwise if the noun is the player:
        issue actor-based library message touching action number 5;
    otherwise:
        issue actor-based library message touching action number 6 for the noun;
    stop the action;
continue the action.

```

Report an actor touching (this is the report touching things rule):

```

if the actor is the player, issue library message touching action number 2
    for the noun;
otherwise issue actor-based library message touching action number 6 for the noun.

```

§117. Waving.

Waving is an action applying to one thing.

The waving action translates into I6 as "Wave".

The specification of the waving action is "Waving in this sense is like waving a sceptre: the item to be waved must be directly held (or worn) by the actor.

In the Standard Rules there are no carry out rules for this action because nothing in the standard Inform world model which reacts to it. The action is provided for authors to hang more interesting behaviour onto for special cases: say, waving a particular rusty iron rod with a star on the end."

§118. Check.

Check an actor waving (this is the can't wave what's not held rule):

if the actor is not the holder of the noun,

stop the action with library message waving action number 1 for the noun.

§119. Report.

Report an actor waving (this is the report waving things rule):

if the actor is the player, issue library message waving action number 2
for the noun;

otherwise issue actor-based library message waving action number 3 for the noun.

§120. Pulling.

Pulling is an action applying to one thing.

The Pulling action translates into I6 as "Pull".

The specification of the pulling action is "Pulling is the act of pulling something not grossly larger than the actor by an amount which would not substantially move it.

In the Standard Rules there are no carry out rules for this action because nothing in the standard Inform world model which reacts to it. The action is provided for authors to hang more interesting behaviour onto for special cases: say, pulling a lever. ('The big red lever is a fixed in place device. Instead of pulling the big red lever, try switching on the lever. Instead of pushing the big red lever, try switching off the lever.')

§121. Check.

Check an actor pulling (this is the can't pull what's fixed in place rule):

if the noun is fixed in place,

stop the action with library message pulling action number 1 for the noun.

Check an actor pulling (this is the can't pull scenery rule):

if the noun is scenery,

stop the action with library message pulling action number 2 for the noun.

Check an actor pulling (this is the can't pull people rule):

if the noun is a person,

stop the action with library message pulling action number 4 for the noun.

§122. Report.

Report an actor pulling (this is the report pulling rule):

if the actor is the player, issue library message pulling action number 3

for the noun;

otherwise issue actor-based library message pulling action number 5 for the noun.

§123. Pushing.

Pushing is an action applying to one thing.

The Pushing action translates into I6 as "Push".

The specification of the pushing action is "Pushing is the act of pushing something not grossly larger than the actor by an amount which would not substantially move it. (See also the pushing it to action, which involves a longer-distance push between rooms.)

In the Standard Rules there are no carry out rules for this action because nothing in the standard Inform world model which reacts to it. The action is provided for authors to hang more interesting behaviour onto for special cases: say, pulling a lever. ('The big red lever is a fixed in place device. Instead of pulling the big red lever, try switching on the lever. Instead of pushing the big red lever, try switching off the lever.')

§124. Check.

Check an actor pushing something (this is the can't push what's fixed in place rule):

if the noun is fixed in place,

stop the action with library message pushing action number 1 for the noun.

Check an actor pushing something (this is the can't push scenery rule):

if the noun is scenery,

stop the action with library message pushing action number 2 for the noun.

Check an actor pushing something (this is the can't push people rule):

if the noun is a person,

stop the action with library message pushing action number 4 for the noun.

§125. Report.

Report an actor pushing something (this is the report pushing rule):

if the actor is the player, issue library message pushing action number 3

for the noun;

otherwise issue actor-based library message pushing action number 6 for the noun.

§126. Turning.

Turning is an action applying to one thing.

The Turning action translates into I6 as "Turn".

The specification of the turning action is "Turning is the act of rotating something - say, a dial.

In the Standard Rules there are no carry out rules for this action because nothing in the standard Inform world model which reacts to it. The action is provided for authors to hang more interesting behaviour onto for special cases: say, turning a capstan."

§127. Check.

Check an actor turning (this is the can't turn what's fixed in place rule):

if the noun is fixed in place,

stop the action with library message turning action number 1 for the noun.

Check an actor turning (this is the can't turn scenery rule):

if the noun is scenery,

stop the action with library message turning action number 2 for the noun.

Check an actor turning (this is the can't turn people rule):

if the noun is a person,

stop the action with library message turning action number 4 for the noun.

§128. Report.

Report an actor turning (this is the report turning rule):

if the actor is the player, issue library message turning action number 3

for the noun;

otherwise issue actor-based library message turning action number 7 for the noun.

§129. Pushing it to.

Pushing it to is an action applying to one thing and one visible thing.
The Pushing it to action translates into I6 as "PushDir".

The specification of the pushing it to action is "This action covers pushing a large object, not being carried, so that the actor pushes it from one room to another: for instance, pushing a bale of hay to the east.

This is rapidly converted into a special form of the going action. If the noun object has the either/or property 'pushable between rooms', then the action is converted to going by the 'standard pushing in directions rule'. If that going action succeeds, then the original pushing it to action stops; it's only if that fails that we run on into the 'block pushing in directions rule', which then puts an end to the matter."

§130. Check.

Check an actor pushing something to (this is the can't push unpushable things rule):
if the noun is not pushable between rooms,
stop the action with library message pushing it to action number 1 for
the noun.

Check an actor pushing something to (this is the can't push to non-directions rule):
if the second noun is not a direction,
stop the action with library message pushing it to action number 2 for
the noun.

Check an actor pushing something to (this is the can't push vertically rule):
if the second noun is up or the second noun is down,
stop the action with library message pushing it to action number 3 for
the noun.

Check an actor pushing something to (this is the standard pushing in directions rule):
convert to special going-with-push action.

Check an actor pushing something to (this is the block pushing in directions rule):
stop the action with library message pushing it to action number 1 for
the noun.

§131. Squeezing.

Squeezing is an action applying to one thing.

The Squeezing action translates into I6 as "Squeeze".

The specification of the squeezing action is "Squeezing is an action which can conveniently vary from squeezing something hand-held, like a washing-up liquid bottle, right up to squeezing a pillar in a bear hug.

In the Standard Rules there are no carry out rules for this action because nothing in the standard Inform world model which reacts to it. The action is provided for authors to hang more interesting behaviour onto for special cases. A mildly fruity message is produced to players who attempt to squeeze people, which is blocked by a check squeezing rule."

§132. Check.

Check an actor squeezing (this is the innuendo about squeezing people rule):

if the noun is a person,

stop the action with library message squeezing action number 1 for
the noun.

§133. Report.

Report an actor squeezing (this is the report squeezing rule):

if the actor is the player, issue library message squeezing action number 2
for the noun;

otherwise issue actor-based library message squeezing action number 3 for the noun.

§134. Saying yes.

Section SR4/8 - Standard actions which always do nothing unless rules intervene

Saying yes is an action applying to nothing.

The Saying yes action translates into I6 as "Yes".

The specification of the saying yes action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§135. Check.

Check an actor saying yes (this is the block saying yes rule):

stop the action with library message saying yes action number 1.

§136. Saying no.

Saying no is an action applying to nothing.

The Saying no action translates into I6 as "No".

The specification of the saying no action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§137. Check.

Check an actor saying no (this is the block saying no rule):

stop the action with library message saying no action number 1.

§138. Burning.

Burning is an action applying to one thing.
The Burning action translates into I6 as "Burn".

The specification of the burning action is
"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§139. Check.

Check an actor burning (this is the block burning rule):
stop the action with library message burning action number 1.

§140. Waking up.

Waking up is an action applying to nothing.
The Waking up action translates into I6 as "Wake".

The specification of the waking up action is
"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§141. Check.

Check an actor waking up (this is the block waking up rule):
stop the action with library message waking up action number 1.

§142. Thinking.

Thinking is an action applying to nothing.

The Thinking action translates into I6 as "Think".

The specification of the thinking action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§143. Check.

Check an actor thinking (this is the block thinking rule):

stop the action with library message thinking action number 1.

§144. Smelling.

Smelling is an action applying to one thing.

The Smelling action translates into I6 as "Smell".

The specification of the smelling action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§145. Check.

Check an actor smelling (this is the block smelling rule):

stop the action with library message smelling action number 1 for the noun.

§146. Listening to.

Listening to is an action applying to one thing.

The Listening to action translates into I6 as "Listen".

The specification of the listening to action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§147. Check.

Check an actor listening (this is the block listening rule):

stop the action with library message listening to action number 1 for the noun.

§148. Tasting.

Tasting is an action applying to one thing.

The Tasting action translates into I6 as "Taste".

The specification of the tasting action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§149. Check.

Check an actor tasting (this is the block tasting rule):

stop the action with library message tasting action number 1 for the noun.

§150. Cutting.

Cutting is an action applying to one thing.

The Cutting action translates into I6 as "Cut".

The specification of the cutting action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§151. Check.

Check an actor cutting (this is the block cutting rule):

stop the action with library message cutting action number 1 for the noun.

§152. Jumping.

Jumping is an action applying to nothing.

The Jumping action translates into I6 as "Jump".

The specification of the jumping action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§153. Check.

Check an actor jumping (this is the block jumping rule):

stop the action with library message jumping action number 1.

§154. Tying it to.

Tying it to is an action applying to two things.
 The Tying it to action translates into I6 as "Tie".

The specification of the tying it to action is
 "The Standard Rules define this action in only a minimal way, blocking it
 with a check rule which stops it in all cases. It exists so that before
 or instead rules can be written to make it do interesting things in special
 cases. (Or to reconstruct the action as something more substantial, unlist
 the block rule and supply carry out and report rules, together perhaps
 with some further check rules.)"

§155. Check.

Check an actor tying something to (this is the block tying rule):
 stop the action with library message tying it to action number 1 for the noun.

§156. Drinking.

Drinking is an action applying to one thing.
 The Drinking action translates into I6 as "Drink".

The specification of the drinking action is
 "The Standard Rules define this action in only a minimal way, blocking it
 with a check rule which stops it in all cases. It exists so that before
 or instead rules can be written to make it do interesting things in special
 cases. (Or to reconstruct the action as something more substantial, unlist
 the block rule and supply carry out and report rules, together perhaps
 with some further check rules.)"

§157. Check.

Check an actor drinking (this is the block drinking rule):
 stop the action with library message drinking action number 1 for the noun.

§158. Saying sorry.

Saying sorry is an action applying to nothing.

The Saying sorry action translates into I6 as "Sorry".

The specification of the saying sorry action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§159. Check.

Check an actor saying sorry (this is the block saying sorry rule):

stop the action with library message saying sorry action number 1.

§160. Swearing obscenely.

Swearing obscenely is an action censored, and applying to nothing.

The Swearing obscenely action translates into I6 as "Strong".

The specification of the swearing obscenely action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§161. Check.

Check an actor swearing obscenely (this is the block swearing obscenely rule):

stop the action with library message swearing obscenely action number 1.

§162. Swearing mildly.

Swearing mildly is an action censored, and applying to nothing.
The Swearing mildly action translates into I6 as "Mild".

The specification of the swearing mildly action is
"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§163. Check.

Check an actor swearing mildly (this is the block swearing mildly rule):
stop the action with library message swearing mildly action number 1.

§164. Swinging.

Swinging is an action applying to one thing.
The Swinging action translates into I6 as "Swing".

The specification of the swinging action is
"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§165. Check.

Check an actor swinging (this is the block swinging rule):
stop the action with library message swinging action number 1 for the noun.

§166. Rubbing.

Rubbing is an action applying to one thing.

The Rubbing action translates into I6 as "Rub".

The specification of the rubbing action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§167. Check.

Check an actor rubbing (this is the block rubbing rule):

stop the action with library message rubbing action number 1 for the noun.

§168. Setting it to.

Setting it to is an action applying to one thing and one topic.

The Setting it to action translates into I6 as "SetTo".

The specification of the setting it to action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§169. Check.

Check an actor setting something to (this is the block setting it to rule):

stop the action with library message setting it to action number 1 for the noun.

§170. Waving hands.

Waving hands is an action applying to nothing.

The Waving hands action translates into I6 as "WaveHands".

The specification of the waving hands action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§171. Check.

Check an actor waving hands (this is the block waving hands rule):

stop the action with library message waving hands action number 1.

§172. Buying.

Buying is an action applying to one thing.

The Buying action translates into I6 as "Buy".

The specification of the buying action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§173. Check.

Check an actor buying (this is the block buying rule):

stop the action with library message buying action number 1 for the noun.

§174. Singing.

Singing is an action applying to nothing.
The Singing action translates into I6 as "Sing".

The specification of the singing action is
"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§175. Check.

Check an actor singing (this is the block singing rule):
stop the action with library message singing action number 1.

§176. Climbing.

Climbing is an action applying to one thing.
The Climbing action translates into I6 as "Climb".

The specification of the climbing action is
"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§177. Check.

Check an actor climbing (this is the block climbing rule):
stop the action with library message climbing action number 1 for the noun.

§178. Sleeping.

Sleeping is an action applying to nothing.

The Sleeping action translates into I6 as "Sleep".

The specification of the sleeping action is

"The Standard Rules define this action in only a minimal way, blocking it with a check rule which stops it in all cases. It exists so that before or instead rules can be written to make it do interesting things in special cases. (Or to reconstruct the action as something more substantial, unlist the block rule and supply carry out and report rules, together perhaps with some further check rules.)"

§179. Check.

Check an actor sleeping (this is the block sleeping rule):

stop the action with library message sleeping action number 1.

§180. Out of world actions. We start with a brace of actions which control the (virtual) hardware of the virtual machine: restore, save, quit, restart, verify, and transcript on and off. All of these are implemented at the I6 level where, in fact, they are delegated quickly to assembly language instructions for whichever is the current VM: so these are close to the metal, as they say.

Section SR4/9 - Standard actions which happen out of world

Quitting the game is an action out of world and applying to nothing.

The quitting the game action translates into I6 as "Quit".

The quit the game rule is listed in the carry out quitting the game rulebook.

The quit the game rule translates into I6 as "QUIT_THE_GAME_R".

Saving the game is an action out of world and applying to nothing.

The saving the game action translates into I6 as "Save".

The save the game rule is listed in the carry out saving the game rulebook.

The save the game rule translates into I6 as "SAVE_THE_GAME_R".

Restoring the game is an action out of world and applying to nothing.

The restoring the game action translates into I6 as "Restore".

The restore the game rule is listed in the carry out restoring the game rulebook.

The restore the game rule translates into I6 as "RESTORE_THE_GAME_R".

Restarting the game is an action out of world and applying to nothing.

The restarting the game action translates into I6 as "Restart".

The restart the game rule is listed in the carry out restarting the game rulebook.

The restart the game rule translates into I6 as "RESTART_THE_GAME_R".

Verifying the story file is an action out of world applying to nothing.

The verifying the story file action translates into I6 as "Verify".

The verify the story file rule is listed in the carry out verifying the story file rulebook.

The verify the story file rule translates into I6 as "VERIFY_THE_STORY_FILE_R".

Switching the story transcript on is an action out of world and applying to nothing.

The switching the story transcript on action translates into I6 as "ScriptOn".

The switch the story transcript on rule is listed in the carry out switching the story transcript on rulebook.

The switch the story transcript on rule translates into I6 as "SWITCH_TRANSCRIPT_ON_R".

Switching the story transcript off is an action out of world and applying to nothing.

The switching the story transcript off action translates into I6 as "ScriptOff".

The switch the story transcript off rule is listed in the carry out switching the story transcript off rulebook.

The switch the story transcript off rule translates into I6 as "SWITCH_TRANSCRIPT_OFF_R".

§181. The VERSION command is not quite so close to the metal – it is implemented in I6, at the end of the day – but it does involve reading the bytes of the story file header, so it needs to take quite different forms for the different formats being compiled to.

Requesting the story file version is an action out of world and applying to nothing.

The requesting the story file version action translates into I6 as "Version".

The announce the story file version rule is listed in the carry out requesting the story file version rulebook.

The announce the story file version rule translates into I6 as "ANNOUNCE_STORY_FILE_VERSION_R".

§182. There's really no very good reason why we provide the out-of-world command SCORE but not (say) TIME, or any one of dozens of other traditional what's-my-status commands: DIAGNOSE, say, or PLACES. But we are conservative on this; it's easy for users or extensions to provide these verbs if they want them, and they are not always appropriate for every project. Even SCORE is questionable, but its removal would be a gesture too far.

Requesting the score is an action out of world and applying to nothing.

The requesting the score action translates into I6 as "Score".

The announce the score rule is listed in the carry out requesting the score rulebook.

The announce the score rule translates into I6 as "ANNOUNCE_SCORE_R".

§183. It's perhaps clumsy to have three actions for switching the style of room description, but this accords with I6 custom (and Infocom's, for that matter), and does no harm.

Preferring abbreviated room descriptions is an action out of world and applying to nothing.

The preferring abbreviated room descriptions action translates into I6 as "LMode3".

The prefer abbreviated room descriptions rule is listed in the carry out preferring abbreviated room descriptions rulebook.

The prefer abbreviated room descriptions rule translates into I6 as "PREFER_ABBREVIATED_R".

The standard report preferring abbreviated room descriptions rule is listed in the report preferring abbreviated room descriptions rulebook.

The standard report preferring abbreviated room descriptions rule translates into I6 as "REP_PREFER_ABBREVIATED_R".

Preferring unabbreviated room descriptions is an action out of world and applying to nothing.

The preferring unabbreviated room descriptions action translates into I6 as "LMode2".

The prefer unabbreviated room descriptions rule is listed in the carry out preferring unabbreviated room descriptions rulebook.

The prefer unabbreviated room descriptions rule translates into I6 as "PREFER_UNABBREVIATED_R".

The standard report preferring unabbreviated room descriptions rule is listed in the report preferring unabbreviated room descriptions rulebook.

The standard report preferring unabbreviated room descriptions rule translates into I6 as "REP_PREFER_UNABBREVIATED_R".

Preferring sometimes abbreviated room descriptions is an action out of world and applying to nothing.

The preferring sometimes abbreviated room descriptions action translates into I6 as "LMode1".

The prefer sometimes abbreviated room descriptions rule is listed in the carry out preferring sometimes abbreviated room descriptions rulebook.

The prefer sometimes abbreviated room descriptions rule translates into I6 as "PREFER_SOMETIMES_ABBREVIATED_R".

The standard report preferring sometimes abbreviated room descriptions rule is listed in the report preferring sometimes abbreviated room descriptions rulebook.

The standard report preferring sometimes abbreviated room descriptions rule translates into I6 as "REP_PREFER_SOMETIMES_ABBR_R".

§184. Similarly, two different actions handle “notify” and “notify off”.

Switching score notification on is an action out of world and applying to nothing.

The switching score notification on action translates into I6 as "NotifyOn".

The switch score notification on rule is listed in the carry out switching score notification on rulebook.

The switch score notification on rule translates into I6 as "SWITCH_SCORE_NOTIFY_ON_R".

The standard report switching score notification on rule is listed in the report switching score notification on rulebook.

The standard report switching score notification on rule translates into I6 as "REP_SWITCH_NOTIFY_ON_R".

Switching score notification off is an action out of world and applying to nothing.

The switching score notification off action translates into I6 as "NotifyOff".

The switch score notification off rule is listed in the carry out switching score notification off rulebook.

The switch score notification off rule translates into I6 as "SWITCH_SCORE_NOTIFY_OFF_R".

The standard report switching score notification off rule is listed in the report switching score notification off rulebook.

The standard report switching score notification off rule translates into I6 as "REP_SWITCH_NOTIFY_OFF_R".

§185. Lastly, the “pronouns” verb, which is perhaps more often used by people debugging the I6 parser than by actual players.

Requesting the pronoun meanings is an action out of world and applying to nothing.

The requesting the pronoun meanings action translates into I6 as "Pronouns".

The announce the pronoun meanings rule is listed in the carry out requesting the pronoun meanings rulebook.

The announce the pronoun meanings rule translates into I6 as "ANNOUNCE_PRONOUN_MEANINGS_R".

§186. **Miscellaneous Grammar Tokens.** There's only one of these, at present.

The understand token a time period translates into I6 as "RELATIVE_TIME_TOKEN".

§187. Grammar.

Section SR4/10 - Grammar

Understand "take [things]" as taking.

Understand "take off [something]" as taking off.

Understand "take [something] off" as taking off.

Understand "take [things inside] from [something]" as removing it from.

Understand "take [things inside] off [something]" as removing it from.

Understand "take inventory" as taking inventory.

Understand the commands "carry" and "hold" as "take".

Understand "get in/on" as entering.

Understand "get out/off/down/up" as exiting.

Understand "get [things]" as taking.

Understand "get in/into/on/onto [something]" as entering.

Understand "get off/down [something]" as getting off.

Understand "get [things inside] from [something]" as removing it from.

Understand "pick up [things]" or "pick [things] up" as taking.

Understand "stand" or "stand up" as exiting.

Understand "stand on [something]" as entering.

Understand "remove [something preferably held]" as taking off.

Understand "remove [things inside] from [something]" as removing it from.

Understand "shed [something preferably held]" as taking off.

Understand the commands "doff" and "disrobe" as "shed".

Understand "wear [something preferably held]" as wearing.

Understand the command "don" as "wear".

Understand "put [other things] in/inside/into [something]" as inserting it into.

Understand "put [other things] on/onto [something]" as putting it on.

Understand "put on [something preferably held]" as wearing.

Understand "put [something preferably held] on" as wearing.

Understand "put down [things preferably held]" or "put [things preferably held] down" as dropping.

Understand "insert [other things] in/into [something]" as inserting it into.

Understand "drop [things preferably held]" as dropping.

Understand "drop [other things] in/into/down [something]" as inserting it into.

Understand "drop [other things] on/onto [something]" as putting it on.

Understand "drop [something preferably held] at/against [something]" as throwing it at.

Understand the commands "throw" and "discard" as "drop".

Understand "give [something preferably held] to [someone]" as giving it to.

Understand "give [someone] [something preferably held]" as giving it to (with nouns reversed).

Understand the commands "pay" and "offer" and "feed" as "give".

Understand "show [someone] [something preferably held]" as showing it to (with nouns reversed).

Understand "show [something preferably held] to [someone]" as showing it to.

Understand the commands "present" and "display" as "show".

Understand "go" as going.

Understand "go [direction]" as going.

Understand "go [something]" as entering.

Understand "go into/in/inside/through [something]" as entering.

Understand the commands "walk" and "run" as "go".

Understand "inventory" as taking inventory.

Understand the commands "i" and "inv" as "inventory".

Understand "look" as looking.

Understand "look at [something]" as examining.
 Understand "look [something]" as examining.
 Understand "look inside/in/into/through [something]" as searching.
 Understand "look under [something]" as looking under.
 Understand "look up [text] in [something]" as consulting it about (with nouns reversed).
 Understand the command "l" as "look".
 Understand "consult [something] on/about [text]" as consulting it about.
 Understand "open [something]" as opening.
 Understand "open [something] with [something preferably held]" as unlocking it with.
 Understand the commands "unwrap", "uncover" as "open".
 Understand "close [something]" as closing.
 Understand "close up [something]" as closing.
 Understand "close off [something]" as switching off.
 Understand the commands "shut" and "cover" as "close".
 Understand "enter" as entering.
 Understand "enter [something]" as entering.
 Understand the command "cross" as "enter".
 Understand "sit on top of [something]" as entering.
 Understand "sit on/in/inside [something]" as entering.
 Understand "exit" as exiting.
 Understand the commands "leave" and "out" as "exit".
 Understand "examine [something]" as examining.
 Understand the commands "x", "watch", "describe" and "check" as "examine".
 Understand "read [something]" as examining.
 Understand "read about [text] in [something]" as consulting it about (with nouns reversed).
 Understand "read [text] in [something]" as consulting it about (with nouns reversed).
 Understand "yes" as saying yes.
 Understand the command "y" as "yes".
 Understand "no" as saying no.
 Understand "sorry" as saying sorry.
 Understand "bother" as swearing mildly.
 ...and so on...

Understand "search [something]" as searching.
 Understand "wave" as waving hands.
 Understand "wave [something]" as waving.
 Understand "set [something] to [text]" as setting it to.
 Understand the command "adjust" as "set".
 Understand "pull [something]" as pulling.
 Understand the command "drag" as "pull".
 Understand "push [something]" as pushing.
 Understand "push [something] [direction]" or "push [something] to [direction]" as pushing it to.
 Understand the commands "move", "shift", "clear" and "press" as "push".
 Understand "turn [something]" as turning.
 Understand "turn [something] on" or "turn on [something]" as switching on.
 Understand "turn [something] off" or "turn off [something]" as switching off.
 Understand the commands "rotate", "twist", "unscrew" and "screw" as "turn".
 Understand "switch [something switched on]" as switching off.
 Understand "switch [something]" or "switch on [something]" or "switch [something] on" as

switching on.

Understand "switch [something] off" or "switch off [something]" as switching off.

Understand "lock [something] with [something preferably held]" as locking it with.

Understand "unlock [something] with [something preferably held]" as unlocking it with.

Understand "attack [something]" as attacking.

Understand the commands "break", "smash", "hit", "fight", "torture", "wreck", "crack", "destroy", "murder", "kill", "punch" and "thump" as "attack".

Understand "wait" as waiting.

Understand the command "z" as "wait".

Understand "answer [text] to [someone]" as answering it that (with nouns reversed).

Understand the commands "say", "shout" and "speak" as "answer".

Understand "tell [someone] about [text]" as telling it about.

Understand "ask [someone] about [text]" as asking it about.

Understand "ask [someone] for [something]" as asking it for.

Understand "eat [something preferably held]" as eating.

Understand "sleep" as sleeping.

Understand the command "nap" as "sleep".

Understand "sing" as singing.

Understand "climb [something]" or "climb up/over [something]" as climbing.

Understand the command "scale" as "climb".

Understand "buy [something]" as buying.

Understand the command "purchase" as "buy".

Understand "squeeze [something]" as squeezing.

Understand the command "squash" as "squeeze".

Understand "swing [something]" or "swing on [something]" as swinging.

Understand "wake" or "wake up" as waking up.

Understand "wake [someone]" or "wake [someone] up" or "wake up [someone]" as waking.

Understand the commands "awake" and "awaken" as "wake".

Understand "kiss [someone]" as kissing.

Understand the commands "embrace" and "hug" as "kiss".

Understand "think" as thinking.

Understand "smell" as smelling.

Understand "smell [something]" as smelling.

Understand the command "sniff" as "smell".

Understand "listen" as listening.

Understand "hear [something]" as listening.

Understand "listen to [something]" as listening.

Understand "taste [something]" as tasting.

Understand "touch [something]" as touching.

Understand the command "feel" as "touch".

Understand "rub [something]" as rubbing.

Understand the commands "shine", "polish", "sweep", "clean", "dust", "wipe" and "scrub" as "rub".

Understand "tie [something] to [something]" as tying it to.

Understand the commands "attach" and "fasten" as "tie".

Understand "burn [something]" as burning.

Understand the command "light" as "burn".

Understand "drink [something]" as drinking.

Understand the commands "swallow" and "sip" as "drink".

Understand "cut [something]" as cutting.

Understand the commands "slice", "prune" and "chop" as "cut".

Understand "jump" as jumping.

Understand the commands "skip" and "hop" as "jump".

Understand "score" as requesting the score.

Understand "quit" or "q" as quitting the game.

Understand "save" as saving the game.

Understand "restart" as restarting the game.

Understand "restore" as restoring the game.

Understand "verify" as verifying the story file.

Understand "version" as requesting the story file version.

Understand "script" or "script on" or "transcript" or "transcript on" as switching the story transcript on.

Understand "script off" or "transcript off" as switching the story transcript off.

Understand "superbrief" or "short" as preferring abbreviated room descriptions.

Understand "verbose" or "long" as preferring unabbreviated room descriptions.

Understand "brief" or "normal" as preferring sometimes abbreviated room descriptions.

Understand "nouns" or "pronouns" as requesting the pronoun meanings.

Understand "notify" or "notify on" as switching score notification on.

Understand "notify off" as switching score notification off.