

# The Inform Designer's Manual

Graham Nelson

*Fourth Edition 2001*

# The Inform Designer's Manual, Fourth Edition

Author: Graham Nelson

Editor: Gareth Rees

Proofreaders: Torbjörn Andersson, Toby Nelson, Andrew Plotkin

Printed edition managed by: David Cornelson

Cover: *Wing of a Roller* (watercolour and gouache on vellum, 1512),  
Albrecht Dürer (1471–1528)

## *Notice of Copyright*

Inform, the program and its source code, its example games and documentation, including the text and typography of this book, are copyright © Graham Nelson 1993, 94, 95, 96, 97, 98, 99, 2000, 01.

The Inform software may be freely distributed provided that: (a) distributed copies are not substantially different from those archived by the author, (b) this and other copyright messages are always retained in full, and (c) no profit is involved. The same conditions apply to distribution or conversion of any electronic form of this book which may be made available by the author. Exceptions to these conditions must be negotiated directly with the author.

A story file produced with the Inform system belongs to whoever wrote it and may be sold for profit if so desired, without the need for royalty payment, provided that it prints a game banner conforming to the standard library's banner at an early stage in play: in particular, this banner must contain the information that the story file was compiled by Inform, and the version numbers of compiler and library used.

The author assumes no liability for errors and omissions in this book, or for damages or loss of revenue resulting from the use of the information contained herein, or the use of any of the software described herein. Inform software is supplied "as is" and carries no warranty, actual or implied.

*First edition September 1994*

*Second expanded edition October 1995*

*Third edition September 1996*

*Web edition with minor revisions May 1997*

*Fourth expanded edition May 2001: Release 4/2 (July 2001)*

*Published by The Interactive Fiction Library (IFLibrary.Org)*

*P.O. Box 3304, St. Charles, Illinois 60174*

*Printed in the United States of America*

*ISBN 0-9713119-0-0*

# Contents

<i>Introduction</i> .....	1
<i>Chapter I: The Inform Language</i>	
§1 <i>Routines</i>	
1.1. Getting started .....	5
1.2. Hello World .....	6
1.3. Routine calls, errors and warnings .....	8
1.4. Numbers and other constants .....	9
1.5. Variables .....	11
1.6. Arithmetic, assignment and bitwise operators .....	12
1.7. Arguments and return values .....	16
1.8. Conditions: if, true and false .....	18
1.9. Code blocks, else and switch .....	21
1.10. while, do... until, for, break, continue .....	24
1.11. How text is printed .....	28
1.12. The print and print_ret statements .....	30
1.13. Other printing statements .....	33
1.14. Generating random numbers .....	34
1.15. Deprecated ways to jump around .....	35
§2 <i>The state of play</i>	
2.1. Directives construct things .....	38
2.2. Constants .....	39
2.3. Global variables .....	39
2.4. Arrays .....	40
2.5. Reading into arrays from the keyboard .....	44
§3 <i>Objects</i>	
3.1. Objects, classes, metaclasses and nothing .....	47
3.2. The object tree .....	49
3.3. Declaring objects 1: setting up the object tree .....	50
3.4. Tree statements: move, remove, objectloop .....	51
3.5. Declaring objects 2: with and provides .....	55
3.6. Declaring objects 3: private properties .....	58
3.7. Declaring objects 4: has and give .....	58
3.8. Declaring objects 5: class inheritance .....	60
3.9. Messages .....	62
3.10. Passing messages up to the superclass .....	64
3.11. Creating and deleting objects during play .....	65
3.12. Sending messages to routines and strings .....	69
3.13. Common properties and Property .....	70
3.14. Philosophy .....	71

*Chapter II: Introduction to Designing*

- §4 *'Ruins' begun* . . . . . 74  
Beginning to lay 'Ruins'; including library files; the `Initialise` routine; some properties of mushrooms; name, description and initial; edible foodstuffs; introducing before and after rules; the stone steps.
- §5 *Introducing messages and classes* . . . . . 82  
Recap of message-sending: a parrot; classes for treasure artifacts: the pygmy statuette, the honeycomb; how clashes are sorted out in class inheritance, additivity.
- §6 *Actions and reactions* . . . . . 87  
Actions are attempts; generating them with `<`, `<<`; the actor, action, noun and second noun; the `##` notation; actions fall into three groups; fake actions like `ThrownAt`; creating new actions, the `Blorple` example; how actions are processed, over 'Before', 'During' and 'After' stages.
- §7 *Infix and the debugging verbs* . . . . . 96  
Three levels; general strategies; command lists and "recording", "replay" and "random" debugging verbs; "showobj", "tree", "showverb", "scope"; keeping watch of "actions", "changes", "messages", "timers"; tracing the parser with "trace"; supernatural ability to "purloin", "abstract", "goto", "gonear"; the Infix ";", ";give", ";move", ";remove", ";examine", ";watch", ";inventory"; marking routines with a `*` to be watched.

*Chapter III: The Model World*

- §8 *Places and scenery* . . . . . 106  
The Square Chamber of 'Ruins'; what "you don't need to refer to"; static and scenery objects; spreading mist with `found_in`;  $\alpha$  *Canis Minoris*; the five senses and reaction rules; rooms have before and after too.
- §9 *Directions and the map* . . . . . 112  
Giving 'Ruins' a small map; `n_to`, `d_to`, etc.; when you `cant_go`; direction objects in the compass; not the same as direction properties.
- §10 *Food and drink* . . . . . 115  
edible foodstuffs; the `Drink` action; difficulties with liquids.
- §11 *Clothing* . . . . . 116  
`Wear` and `Disrobe`; wearable items of clothing; a jade face mask.
- §12 *Containers, supporters and sub-objects* . . . . . 118  
Containers: `container`, `supporter`, `capacity`, `open`, `openable`; locks and keys: `locked`, `lockable`, `with_key`; `LetGo` and `Receive` to trap use of a container: a chasm; transparency and component parts.

§13	<i>Doors</i> . . . . .	122
	How to create a door; <code>door_to</code> , <code>door_dir</code> ; <code>when_open</code> , <code>when_closed</code> ; a stone door for ‘Ruins’; a two-way door, the ‘Advent’ grate; why <code>door_dir</code> is needed and how to trap every attempt to go through.	
§14	<i>Switchable objects</i> . . . . .	125
	<code>switchable</code> and <code>on</code> : <code>when_on</code> , <code>when_off</code> ; the Gotham City searchlight; a sodium lamp; describe taking precedence.	
§15	<i>Things to enter, travel in and push around</i> . . . . .	127
	enterable objects: a slab altar; vehicles: KAR 1; special rule about the Go action when inside something enterable; the PushDir action: a huge pumice-stone ball; pushing up and down.	
§16	<i>Reading matter and consultation</i> . . . . .	131
	The Consult action, “look up”; <code>consult_from</code> and <code>consult_words</code> : a dictionary of glyphs, Tyndale’s Bible; making “read” and “examine” different.	
§17	<i>People and animals</i> . . . . .	134
	<code>animate</code> objects and the <code>life</code> rule; a coiled snake and a mummified priest; Blofeld’s reactions; talkable objects; some people are transparent.	
§18	<i>Making conversation</i> . . . . .	138
	Orders are actions for other people; giving people their own grammars; untypeable verbs; several voice-activated machines; fake fake actions; telephony.	
§19	<i>The light and the dark</i> . . . . .	144
	Light versus darkness is automatic; modifying the darkness; going from dark into dark and the DarkToDark entry point; rules on ‘when there is light’.	
§20	<i>Daemons and the passing of time</i> . . . . .	147
	Daemons and the <code>daemon</code> property; starting and stopping them; background daemons; timers (fuses); <code>time_left</code> and <code>time_out</code> ; <code>each_turn</code> events for places and nearby objects; the time of day; changing it with <code>SetTime</code> ; on the status line; midnight, sunrise, sunset; the exact sequence of events at end-of-turn.	
§21	<i>Starting, moving, changing and killing the player</i> . . . . .	152
	What <code>Initialise</code> should do; the <code>location</code> ; initial restoration; teleportation and the <code>PlayerTo</code> routine; what happens when the room changes: <code>NewRoom</code> , <code>initial</code> for a room, <code>visited</code> ; giving the player his own <code>before</code> rule; using <code>ChangePlayer</code> to transform him into any object; multi-character games; life and <code>deadflag</code> ; the <code>DeathMessage</code> routine; resurrection and the <code>AfterLife</code> .	
§22	<i>Miscellaneous constants, scoring, quotations</i> . . . . .	160
	<code>Story</code> and <code>Headline</code> ; <code>MAX_CARRIED</code> ; the automatic “sack object”; ‘amusing’ rewards for the victorious; two scoring systems: for places and items, or for completing tasks; rankings and <code>PrintRank</code> ; automatic score notification and <code>notify_mode</code> ; “objects” and “places” verbs, removable with <code>NO_PLACES</code> ; boxed quotations.	

§23	<i>'Ruins' revisited</i> . . . . .	166
	Further examples to complete 'Ruins'; a map and a step by step solution of the final game.	
§24	<i>The world model described</i> . . . . .	176
	¶1. Substance – ¶2. Containment – ¶3. Space – ¶4. Sense – ¶5. Time – ¶6. Action.	
§25	<i>Extending and redefining the world model</i> . . . . .	188
	Enriching the model; Egyptian amulets and their spells; making a new library file; the LibraryMessages system for changing messages like "Dropped."; changing the prompt; the last resort of Replace directives; using the world model description; fire and flammability.	
 <i>Chapter IV: Describing and Parsing</i>		
§26	<i>Describing objects and rooms</i> . . . . .	194
	print (The) obj, . . . (the) obj and so on; indefinite and definite article; proper nouns; the short_name of an object; invent strings and routines; exactly how inventory lines are printed; a matchbook; describe routines; exactly how rooms are described; Locale.	
§27	<i>Listing and grouping objects</i> . . . . .	201
	The list-maker WriteListFrom; its style bitmap; examples: tall and wide inventories; grouping similar items together in lists: foodstuffs, Scrabble pieces and denominations of coin.	
§28	<i>How nouns are parsed</i> . . . . .	206
	How name is used; a fried green tomato turning red; the parser breaks text into a stream of words; wn and NextWord; reading words as numbers or from their raw text; a parse_name routine is much more flexible than name; the ParseNoun entry point; distinguishing adjectives from nouns.	
§29	<i>Plural names for duplicated objects</i> . . . . .	212
	Collections of indistinguishable objects; a bag of six coins; the plural property for printing out plurals; definition of 'indistinguishable'; writing parse_name routines to allow plurals to be understood; class of crowns.	
§30	<i>How verbs are parsed</i> . . . . .	215
	The parser's fundamental method; BeforeParsing entry point; the actor and verb word; synonyms for verbs; definitions of grammar, line and token; action_to_be; Verb directive: a simplified "take" grammar; meta verbs; grammar creates actions; creating an "xyzy" verb; how to Extend grammar for an existing verb: pushing numbered buttons; priority: replace, first, last; splitting synonymous verbs apart with Extend only; the UnknownVerb and PrintVerb entry points.	

- §31 *Tokens of grammar* . . . . . 222  
 Full list of grammar tokens; prepositions; noun and held; implicit taking; tokens allowing multiple objects like “all”; filtering out nouns by attribute: “use” verb; and by general routine: “free” verb; parsing numbers: “type” verb, ParseNumber; general parsing routines; reading from the parser’s raw text buffer and parse table; exercises, including French, telephone and floating-point numbers, times of day, adding a third parameter to a grammar line.
- §32 *Scope and what you can see* . . . . . 231  
 The definition of ‘in scope’; touchability is stricter than scope; answering questions: “what is a grue”; scope=. . . tokens with programmable scope; scope\_stage, ScopeWithin and PlaceInScope; changing the global definition of ‘in scope’ using InScope; scope\_reason; looping over and testing scope; making the rules more sensitive to darkness; a long room divided by a glass wall; the add\_to\_scope property for component parts of containers.
- §33 *Helping the parser out of trouble* . . . . . 237  
 Parser error messages and ParserError; ambiguity-resolution and influencing it with ChooseObjects; making “eat” prefer edible objects; redefining “all”; exactly how ambiguities are resolved.
- Chapter V: Natural Language*
- §34 *Linguistics and the Inform parser* . . . . . 243  
 ¡Bienvenido a Aventura!; Informese; commands, verb phrases, prepositions, noun phrases, gender-number-animation (GNA), descriptors, nouns, pronouns; example of parsing; grammatical features not present in Informese.
- §35 *Case and parsing noun phrases* . . . . . 252  
 Flexion and cases; parsing inflected noun phrases; for example, Old English dative word-endings.
- §36 *Parsing non-English languages* . . . . . 257  
 Compromises; accented characters at the keyboard?; dialects; language definition files; Zcharacter and the customisation of ZSCII; specifying pronouns and descriptors; translating natural languages to Informese.
- §37 *Names and messages in non-English languages* . . . . . 269  
 Gender-number-animation of short names; agreement with articles and cases; contraction forms and articles; library messages of all kinds.

*Chapter VI: Using the Compiler*

- §38 *Controlling compilation from within* . . . . . 278  
 Include; conditional compilation: If . . . , Ifnot, Endif; Message; linking in the library; writing new modules to link in; serial and release numbers.
- §39 *Controlling compilation from without* . . . . . 285  
 Switches; memory settings; pathname variables; Inform Control Language (ICL).
- §40 *All the Inform error messages* . . . . . 290  
 Fatal errors; errors, including linker and assembler errors; warnings, including obsolete usage warnings.

*Chapter VII: The Z-Machine*

- §41 *Architecture and assembly* . . . . . 304  
 ZIL, Inform and the Z-machine; its Versions; assembly language; store and branch opcodes, labels; memory map and stack.
- §42 *Devices and opcodes* . . . . . 308  
 The Standard for interpreters; input and output streams; the Version 5 and Version 6 screen models; colours, windows and status lines; interrupt countdowns; pictures; sounds; keyboard reading with timed interrupts; terminating characters; the mouse; menus; loading and saving auxiliary files; throwing and catching stack frames.
- §43 *Pictures, sounds, blurbs and Blorb* . . . . . 321  
 What a Blorb file is; the perlBlorb utility program; blurb files; palette, resolution, scaling; specifying what sounds and pictures belong to a story file.
- §44 *Case study: a library file for menus* . . . . . 326  
 Invisiclues; basic design; the Option class; the Menu class; the SwitchOption class.
- §45 *Limitations and getting around them* . . . . . 334  
 Story file size; readable memory size; grammar; vocabulary; dictionary resolution; objects, attributes, properties, names; function and message arguments; recursion and stack usage; abbreviations in text.



*Chapter VIII: The Craft of Adventure*

§46	<i>A brief history of interactive fiction</i>	
§46.1.	Precursors and university games 1972–81	342
§46.2.	The commercial boom 1982–6	349
§46.3.	The growth of a community c. 1985–91	357
§46.4.	Newsgroups and revival 1992–	360
§47	<i>Realities and origins</i>	364
	Fictional backgrounds; crimes against mimesis; historical research; book adaptations.	
§48	<i>A triangle of identities</i>	368
	Player, protagonist and narrator; participatory magic; the overture and other narrated text.	
§49	<i>Structure</i>	375
	Size and density; the prologue, middle game and end game; wide versus narrow; lattice diagrams, including one for ‘Ruins’.	
§50	<i>The design of puzzles</i>	382
	General remarks on good and bad puzzles; mazes, light sources, capacity and exhaustion, timed puzzles, utility objects, keys and doors, machinery and vehicles, fire, water, air, earth, plants, animals, monsters, people, ropes and chains, riddles, decipherment puzzles; clues, luck and accidental solutions; optional and multiple solutions; rewards.	
§51	<i>The room description</i>	396
	Examples of good, mediocre and poor description; how much space is one location?; variable descriptions; outdoor games; differing perspectives.	
§52	<i>Finishing</i>	402
	Scoring systems; responding to wrong guesses; examples of typical bugs; play-testing and editing; concluding homily.	

*APPENDICES*

§A1	<i>Library attributes</i>	408
§A2	<i>Library properties</i>	412
§A3	<i>Library routines</i>	419
§A4	<i>Library message numbers</i>	426
§A5	<i>Entry point routines</i>	431
§A6	<i>Answers to all the exercises</i>	435

*TABLES*

*Table 1 Operators* . . . . . 517  
*Table 2 The ZSCII Character Set* . . . . . 519  
*Table 3 Command Line Switches* . . . . . 521  
*Table 4 Statements* . . . . . 522  
*Table 5 Directives* . . . . . 523  
*Table 6 Library actions* . . . . . 524

*INDEX*

*Cited Works of Interactive Fiction* . . . . . 528  
*Index of Exercises* . . . . . 534  
*General Index* . . . . . 536