

# Глава 1. Что нужно подготовить для работы с книгой

## 1.1. Как организовать файловую структуру системы LevInfinite для работы в вычислительной среде MatLab

Естественно, на компьютере, прежде всего, должна быть установлена и освоена версия 6.x MatLab или старше. Замечательно отношение фирмы MatWorks к совместимости различных выпусков программы MatLab. Если вы напишете какой-то программный файл на языке MatLab, то вы можете быть уверены, что написали «вечный» текст. Я пользуюсь программой MatLab уже 25 лет, с момента выхода ее первой версии, когда ее дистрибутив помещался на одной дискете объемом 1.4 Мб. И все тексты моих m-скриптов и функций, написанные в начале славных дел, работают до сих пор, если я их запущу на исполнение в современной версии MatLab. Ни одна другая система программирования не обладает таким замечательным свойством. Если вы программировали на Фортране 20 лет назад, то современные трансляторы могут не понять ваш древний программный текст, наткнувшись на так называемые устаревшие средства Фортрана. В системе MatLab такая ситуация немыслима.

Я настаиваю на версии 6.x MatLab только потому, что я написал некоторые m-программы во вспомогательном средстве визуального программирования GUIDE, позволяющем создать современный пользовательский интерфейс управления вычислениями. Я знаю, что GUIDE в версии 6.x отличается от версии 5. Поэтому я осторожничаю.

Впрочем, квалифицированный пользователь среды MatLab может извлечь все функции из m-файла моей программы, написанной с помощью GUIDE, и превратить эти функции в независимые от GUIDE вычислительные средства. Такие независимые функции, как я подозреваю, будут «вечными». Однако, удобный графический пользовательский интерфейс, создаваемый с помощью GUIDE, является серьезным соблазном, и я его не избежал.

Далее, надо приготовить файловую систему заранее определенной структуры. Вся эта структура называется LevInfinite по имени своей главной папки. В этом названии ‘Lev’ есть аббревиатура содержательного названия ‘Light-handed Elucidator of Vibrations’. С другой стороны, это имя автора теории, заложенной в вычислительные алгоритмы – Lev A. Gribov. Определение ‘Infinite’ отражает наш давний восторг, когда нам впервые удалось создать систему программ, не скованных ограниченными размерами модели из-за скудной оперативной памяти старых вычислительных машин. Теперь динамическим распределением памяти никого не удивишь. А MatLab, среди других языков программирования, чемпион по динамическим массивам. Так что вторая часть названия это дань прошлому.

В книге принято, что вся структура LevInfinite расположена в корне жесткого диска C:. Конечно, операционная система Windows позволяет расположить файлы и как-то иначе. Как это сделать, будет ясно из описания самой структуры. На компакт-диске, прилагаемом к книге, имеется готовая файловая структура LevInfinite, которую можно просто скопировать на диск C:.

Далее, надо настроить MatLab так, чтобы папка Programs с программами LevInfinite была назначена в качестве рабочей директории MatLab (Current Directory в главном окне).

Надо проследить еще за тем, чтобы в системных установках Windows в качестве разделителя целой и десятичной части чисел была задана десятичная точка, а не запятая, как принято в русифицированных ПК. К этой установке можно добраться, выполнив на ПК цепочку команд: **Пуск – Настройка – Панель управления – Язык и стандарты – вкладка Числа** – в окне *Разделитель целой и дробной части числа* установить точку.

После таких установочных работ можно приступать к ознакомлению с содержимым папок LevInfinite, затем к изучению тестового примера либо знакомству с инструкциями к отдельным программам.

Осмотр папок LevInfinite позволит пользователю ознакомиться со следующими особенностями нашей системы визуального молекулярного моделирования.

## **Обзор содержимого папок комплекса LevInfinite**

Вся совокупность папок и файлов LevInfinite находится в папке **LevInfinite**. В папке **LevInfinite** находятся следующие вложенные папки.

**Config**  
**Molecules**  
**Programs**

Содержание этих папок таково.

**Config.** Здесь находятся три файла:

**Molecules.ini.** Это текстовый файл с единственной записью, определяющей полный путь к папке Molecules. Если, например, **LevInfinite** находится на диске C:, то запись в данном файле должна быть **C:\LevInfinite\Molecules\**. При помещении папки **LevInfinite** на другой диск эту запись следует отредактировать вручную.

**ToCalculate.ini.** Это текстовый файл с единственной записью, определяющей имя той молекулы, с которой имеют дело все вычислительные программы в данном сеансе редактирования модели и расчёта её колебательного спектра. С точки зрения операционной системы это имя является именем папки, содержащей все файлы, относящиеся к молекуле с данным именем. Если, например, в данном файле содержится запись **butan**, то все вычислительные программы **LevInfinite** предполагают, что надо иметь дело с данными, хранящимися в папке **C:\LevInfinite\Molecules\butan**. Запись в файле **ToCalculate.ini** создаётся автоматически самими программами, формирующими или редактирующими молекулярную модель. Если же надо перейти к

вычислениям для иной существующей модели, то содержимое этого файла надо изменить вручную с помощью любого текстового редактора.

**ToJoin.txt** используется программой визуального формирования крупной молекулярной модели из фрагментов двух ранее созданных моделей. Это текстовый файл с информацией об именах сшиваемых моделей, о номерах атомов, образующих накладываемые друг на друга связи, и о способе взаимной ориентации сшиваемых фрагментов новой модели.

**Molecules.** Здесь содержатся вложенные папки с данными о молекулярных моделях. Каждой модели выделяется отдельная папка, содержащая все файлы с результатами расчётов и файлы \*.mat с полной информацией о параметрах молекулярной модели. Папки могут иметь смысловые имена, совпадающие с химическими названиями молекул. Папок может быть сколько угодно. Для фрагментного расчета обязательна папка **Joint**.

**Programs.** Здесь хранятся все программы комплекса.

Рассмотрим содержание папки, где хранятся все данные о молекулярной модели. Например, откроем папку EthaneML, где имеется вся информация о колебательной модели этана. 'ML' в названии папки напоминает, что модель приготовлена в среде MatLab. Мы увидим следующие имена файлов.

**Bstruct.mat.** Это файл, где хранятся все параметры модели этана, описывающие структуру матрицы **B** этой модели и сценарии вычисления элементов матрицы **B**. Файл доступен только программам, написанным на языке MatLab. Его нельзя прочесть с помощью обычного текстового редактора.

**Umiu.mat.** Это файл, где хранятся все силовые и электрооптические параметры модели этана. Здесь хранится также информация, необходимая для программного распознавания характерных взаимодействий колебательных координат модели. Файл доступен только программам, написанным на языке MatLab. Его нельзя прочесть с помощью обычного текстового редактора.

**AtomCoords.txt.** Это декартовы координаты атомов в лабораторной системе координат.

**AtomMasses.txt.** Это массы атомов.

**AtomSymbols.txt.** Это химические символы атомов.

**BqStructures.txt.** Это файл, описывающий колебательные координаты  $q$  типа растяжения связи. Для каждой координаты типа  $q$  приведены номера двух атомов, образующих связь. Нумерация атомов соответствует порядку их следования в файле AtomCoords.txt. Для каждого атома приведены тройки компонент соответствующих векторных элементов матрицы **B**.

**BondLengths.txt.** Это длины связей в порядке их упоминания в списке координат типа  $q$ .

**BondVectors.txt.** Это компоненты единичных векторов связей в порядке их упоминания в списке координат типа  $q$ .

**BaStructures.txt.** Это файл, описывающий колебательные координаты  $\alpha$  типа изменения валентного угла. Для каждой координаты типа  $\alpha$  приведены номера трех атомов, образующих валентный угол. Атом, находящийся в вершине угла,

упомянут вторым. Для каждого атома приведены тройки компонент соответствующих векторных элементов матрицы **B**.

**BgStructures.txt**. Это файл, описывающий колебательные координаты  $\gamma$  типа изменения валентного угла на линейном участке модели. Для каждой координаты типа  $\gamma$  приведены номера трех атомов, образующих валентный угол. Атом, находящийся в вершине угла, упомянут вторым. Для каждого атома приведены тройки компонент соответствующих векторных элементов матрицы **B**.

**BrStructures.txt**. Это файл, описывающий колебательные координаты  $\rho$  типа изменения угла между валентной связью и плоскостью, образованной двумя другими валентными связями. При этом имеется общий атом у тройки атомов, образующих плоскость, и у связи, составляющей пространственный угол с плоскостью. Этот атом указан вторым. Свободный атом связи упомянут четвертым. Для каждого атома приведены тройки компонент соответствующих векторных элементов матрицы **B**.

**BhStructures.txt**. Это файл, описывающий колебательные координаты  $\chi$  типа изменения угла между двумя плоскостями модели. Плоскости задаются тройками атомов. Для каждого атома приведены тройки компонент соответствующих векторных элементов матрицы **B**.

**Symmetry.txt** и **Cmatrix.txt**. Для симметричной модели в этих двух файлах даны описания группы симметрии модели и структуры матрицы преобразования симметрии.

**UqSparse.txt**. Здесь в форме верхнего треугольника редкой квадратной матрицы **U** записана информация о силовом поле модели.

**Ts.txt** и **Us.txt**. Для симметричной модели в этих двух файлах по блокам симметрии записаны полные верхние треугольники матриц кинематических и силовых коэффициентов. Для несимметричной модели вместо них мы найдем файлы **Tp.txt** и **Uq.txt**.

**Bdr.txt** и **Tau.txt**. Это результаты диагонализации матрицы кинематических коэффициентов, записанные по блокам симметрии.

**Ms.txt**. Здесь по блокам симметрии записано, сколько ненулевых собственных чисел нашлось после диагонализации матрицы кинематических коэффициентов.

**Freqs.txt**. Это рассчитанные частоты колебаний модели, записанные по блокам симметрии.

**Lp.txt**, **Lq.txt**, **Ls.txt**, **Ps.txt**, **Shifts.txt**. Это различные представления форм колебаний модели. Данные записаны по блокам симметрии.

**Miu.txt**, **dmdqSparse.txt**, **sigma.txt**. Здесь хранятся электрооптические параметры модели.

**IRSabsint.txt**. Это результаты расчета абсолютных интенсивностей ИК поглощения и скоростей изменения дипольного момента по осям декартовых координат в нормальных колебаниях модели.

**InfTable.txt**. Здесь содержится наиболее общая информация о модели для расчетных программ. Приведены значения таких величин:  $N_{atoms}$  – число атомов;  $N_q$  – число координат типа  $q$ ;  $N_\alpha$  – число координат типа  $\alpha$ ;  $N_\gamma$  – число координат типа  $\gamma$ ;  $N_\rho$  – число координат типа  $\rho$ ;  $N_\chi$  – число координат типа  $\chi$ ;  $N$  – общее число колебательных координат;  $N_{bs}$  – число блоков симметрии,  $M_f$  – число нормальных колебаний.

**Errors.txt**. Здесь находится описание фатальной ошибки, если она возникла в какой-то из расчетных программ и привела к остановке программы.

Некоторые вычислительные программы создают или вводят свои специфические файлы, о которых сказано в инструкциях к этим программам.

Все упомянутые текстовые файлы можно очень легко прочесть и, при необходимости, отредактировать с помощью любого текстового редактора.

#### Назначение программ, находящихся в папке **Programs**

<b>Программа</b>	<b>Назначение</b>	<b>Вводит</b>	<b>Выводит</b>
Vstruct.m (работает в паре с файлом Vstruct.fig)	Создаёт новую модель или редактирует ранее созданную в плане геометрии молекулы и системы её колебательных координат. Тем самым, подготавливает в визуальном режиме всю информацию о структуре матрицы <b>V</b> колебательной молекулярной модели.	В диалоге с пользователем вводит информацию о положениях атомов в пространстве, о связях между атомами в каждом типе колебательных координат. Либо импортирует информацию об атомах и связях из системы текстовых файлов.	Выводит данные о модели в папку с именем молекулы. Основные данные о модели для дальнейшего использования выводятся в файл Vstruct.mat. Данные для расчётных программ выводятся в виде текстовых файлов
JoinFragments	Создаёт новую модель из двух готовых моделей, соединяя их через валентную связь.	Вводит текстовый файл с информацией об именах соединяемых моделей и о способе их сшивки.	Выводит данные о модели в папку Joint для новой модели в виде файлов, подобно программе Vstruct.m
Umiu.m (работает в паре с файлом Umiu.fig)	Создаёт или редактирует систему силовых постоянных и электрооптических параметров модели	Вводит информацию о системе колебательных координат из файла Vstruct.mat и информацию о новых или изменяемых значениях параметров. Программа способна импортировать силовые	Выводит значения параметров в файл Umiu.mat. Данные для расчётных программ выводятся в виде текстовых файлов

		параметры из ранее приготовленной модели.	
Symlev.exe	Находит группу симметрии молекулы, формирует матрицу симметрии молекулы.	Вводит текстовые файлы с информацией о структурах групп колебательных координат модели.	Выводит текстовые файлы с информацией о структуре матрицы симметрии.
MatrixT.m	Формирует матрицу $T_p$ для несимметричной модели или матрицу $T_s$ для симметричной.	Вводит текстовые файлы с информацией о структурах групп колебательных координат модели и учитывает симметрию.	Выводит Текстовый файл с матрицей $T_p$ или $T_s$ для симметричной модели.
Diag1.m (Diag1.exe для очень крупных моделей)	Находит число нормальных колебаний молекулы, решает задачу о собственных значениях и собственных векторах матрицы $T_s$ ( $T_p$ ).	Вводит текстовый файл с матрицей $T_s$ ( $T_p$ ).	Выводит текстовые файлы с информацией о собственных значениях и собственных векторах матрицы $T_s$ ( $T_p$ ).
MatrixU.m	Формирует матрицу $U_q$ , вычисляет матрицу $U_s$ для симметричной модели.	Вводит текстовые файлы с информацией о структуре матрицы $U_q$ и учитывает симметрию.	Выводит текстовый файл с матрицей $U_s$ ( $U_q$ ).
Diag2.m (Diag2.exe для очень крупных моделей)	Вычисляет частоты, формы колебаний и смещения атомов молекулы	Вводит текстовый файл с матрицей $U_s$ ( $U_q$ ), текстовые файлы с информацией о собственных значениях и собственных векторах матрицы $T_s$ ( $T_p$ ), текстовые файлы с информацией о структурах групп колебательных координат модели,	Выводит текстовые файлы с информацией о частотах, формах колебаний и смещениях атомов молекулы

		текстовые файлы с информацией о структуре матрицы симметрии	
intir.m	Вычисляет абсолютные интенсивности в ИК спектре поглощения модели	Вводит информацию об электрооптических параметрах и о формах колебаний модели	Выводит файл IRSabsint.txt
IRcurve.m	Строит спектральную кривую ИК поглощения модели	Вводит файлы IRSabsint.txt и sigma.txt	
MolView.exe	Визуализует колебания молекулы	Вводит файлы с информацией о геометрии, частотах, формах колебаний и смещениях атомов молекулы.	

## 1.2. Файловая структура численного представления модели глазами разработчика программ

В этом разделе разработчик приведенных в книге программ делится опытом с потенциальными разработчиками новых версий данного комплекса.

**В процессе разработки этого комплекса программ было принято решение:** все программы визуального моделирования колебаний молекул должны обмениваться друг с другом информацией через файлы самой простой структуры, то есть через текстовые файлы последовательного доступа. При этом каждая логически самостоятельная единица передаваемой информации должна содержаться в отдельном файле. Конечно, таких файлов будет довольно много, что затрудняет идентификацию модели и ставит под угрозу целостность информации о модели. Но в нашем распоряжении имеются технические средства для обхода этого затруднения. Будем идентифицировать данную модель именем папки, в которой и будут храниться все многочисленные файлы, относящиеся к данной модели. Папка может нести осмысленное название, например, Propylene. А все файлы внутри этой папки могут иметь содержательные, но уже стандартные имена, например AtomMasses, AtomSymbols, Tr, Uq, Lp, Lq и так далее. Если папку целиком копировать, перемещать, уничтожать, то целостность данных будет обеспечена самой операционной системой.

Любой текстовый файл последовательного доступа легко прочесть с помощью доступного текстового редактора. Следовательно, пользователь в любой момент может заглянуть в файл и увидеть, что собой представляет данная информация. Ещё важнее, что весь программный комплекс приобретает

небывалую доселе гибкость как инструмент исследовательской работы. Если пользователь в процессе моделирования вдруг захочет изменить массу какого-то атома, то ему не нужно будет проходить весь длинный путь создания молекулярной модели. Достаточно с помощью текстового редактора заменить нужную массу в файле AtomMasses.txt и затем обратиться к программам, которые, так или иначе, используют значения масс атомов. Никаких специальных утилит для чтения и правки промежуточных данных нам не понадобится. Они уже есть в любой операционной системе. Нужно только помнить, что такая свобода допускается лишь для промежуточных результатов вычислений, когда надо просмотреть несколько вариантов расчета при вариации некоторых параметров модели. Наряду с такими легко редактируемыми файлами есть такие, которые предназначены для длительного хранения готовой модели. В таких файлах изменять параметры с помощью текстового редактора нельзя. Это при необходимости можно также просто делать с помощью основных программ комплекса LevInfinite, поскольку данные программы не только создают модели, но и с легкостью их редактируют.

Немаловажно, что такой формат промежуточных данных позволяет нам совершенно свободно ставить части нашего программного комплекса на машины с различными операционными системами, чтобы задействовать объединённые вычислительные мощности машин, связанных сетью. Все коммуникационные средства легче всего оперируют с информацией, передаваемой из системы в систему в виде простых текстовых файлов. Это, например, позволяет нам перекачивать информацию от рабочей станции на суперкомпьютер для выполнения громоздкого расчета и назад без написания утилит интерфейса. Все нужные нам утилиты уже содержатся на обоих концах линии связи.

Данное решение о формате файлов является не совсем удачным для накопления банка данных о молекулярных моделях. Поэтому **дополняем принятое решение следующим**: промежуточные расчёты оформляются в виде набора текстовых файлов только на время работы над моделью в процессе исследования. Для длительного хранения **модель оформляется в виде файлов \*.mat** системы MatLab. Как известно, файлы с расширением .mat создаются командой **save** для сохранения данных из рабочей области (Workspace) системы MatLab. Работая с любой программой визуального моделирования из данной книги, всегда можно закрепить успех, даже промежуточный, отправив снимок рабочей области в файл с расширением .mat. Затем можно вернуться к этой же программе, восстановить состояние рабочей области с помощью команды **load** и продолжать работу над той же моделью. Для законченной модели файлы .mat послужат удобным и компактным хранилищем всей основной информации о модели. Текстовые файлы можно не сохранять. А при необходимости, можно восстановить все текстовые файлы модели с помощью тех же программ, с помощью которых эта модель создавалась. Для этого все крупные программы, содержащие встроенную команду **save**, содержат и команды для развертывания файлов .mat в текстовые файлы.

## Структура программного комплекса

Откажемся от принятой в LEV DOS и в LEV Windows технической идеи строить комплекс как гигантскую оболочку, обнимающую всю функциональность вычислительных программ. Откажемся от принятой в LEV

Windows технической идеи строить одну единственную гигантскую вычислительную программу, решающую все задачи анализа колебаний.

Опыт показал, что гигантская оболочка не способна охватить всю функциональность, необходимую в исследовательской работе. Довольно быстро после выпуска LEV DOS к нему стали добавлять отдельно стоящие утилиты для вычисления новых функций. Гигантскую оболочку или всеохватывающую вычислительную программу трудно развивать и сопровождать.

Вернёмся к структуре комплекса, принятой в давно устаревших версиях LEV ЕСЭВМ и LEV 100 для персональных компьютеров. То есть каждая законченная вычислительная операция по формированию и преобразованию молекулярной колебательной модели выполняется собственной программой. Например, MatrixT, MatrixU, Diag1, Diag2. Вызывать вручную все программы из такой цепочки утомительно. Но учтём, что современные средства визуального программирования (прежде всего, GUIDE в версии 6.x MatLab) позволяют за 5 минут приготовить управляющую программу, которая автоматически вызовет все программы по цепочке. А цепочку может заказать самому себе или программисту автор исследования. Также легко запрограммировать некий стандартный сценарий последовательного вызова вычислительных программ и предусмотреть в этом сценарии кнопочное управление выбором различных опций хода расчёта. На программирование такого сценария уйдёт 10 минут.

Преимущество такой раздробленной структуры программного комплекса особенно сказывается при выходе с нашими задачами на суперкомпьютер. Головные программы, формирующие колебательное уравнение, перенасыщены логикой и требуют значительного участия пользователя. Это принципиально медленные программы, которые удобно выполнять в диалоговом режиме на рабочей станции. А программы расчёта колебаний и решения обратных задач просты по логике, но требуют значительных вычислительных ресурсов. Поэтому их алгоритмы реализованы на двух языках – на MatLab и на Fortran-90. Последние можно поставить на суперкомпьютер вместе с подготовленными для них на рабочей станции исходными данными. Предлагаемая структура программного комплекса позволяет поступать так без какой-либо переделки вычислительных программ. Просто надо оттранслировать их и на рабочих станциях, и на суперкомпьютере.

### **Типы программ в системе LevInfinite и необходимые языки программирования**

1. Программы, обеспечивающие визуальное конструирование и модификацию молекулярной модели. Эти программы основаны на пользовательском интерфейсе Windows, созданном с помощью GUIDE в версии 6.x MatLab. Отличие этих программ состоит в том, что они постоянно держат связь с файлами формата .mat, хранящими создаваемую или модифицируемую молекулярную модель в форме снимков рабочей области MatLab.. Это позволяет в любой момент приостановить формирование модели и сохранить уже имеющуюся часть модели. В любой момент можно продолжить наращивание модели. В любой момент можно вернуться к любому структурному элементу модели и модифицировать его параметры, не проходя с начала до конца длинный диалоговый путь к этому элементу. Программы обеспечат полную функциональную свободу формирования любых элементов модели: добавить атом, удалить атом, изменить координаты любого атома, даже входящего в циклы, повернуть в пространстве любую группу атомов как целое,

передвинуть в пространстве любую группу атомов как целое. Для таких программ подходящей средой программирования является GUIDE в версии 6.x MatLab. Трудоёмкость написания современного пользовательского интерфейса получается минимальной при высоком качестве самого интерфейса. Такие программы очень легко модифицировать и развивать. Недостатком является только перегруженность кода командами и параметрами графического интерфейса. Это затрудняет выделение тех фрагментов кода, которые относятся непосредственно к вычислениям матричных элементов модели. Впрочем, великолепная организация структуры кода, создаваемого средствами GUIDE, позволяет опытному программисту обходить эти затруднения.

2. Программы, вычисляющие специфические функции для молекулярной колебательной модели. Эти программы не содержат никакого пользовательского интерфейса. Они вводят файлы, подготовленные программами типа 1, и выводят файлы для продолжения счёта другими программами типа 2. Некоторые промежуточные результаты передаются программам типа 1 для визуального анализа результатов. Данные программы надо разрабатывать в расчете как на рабочие станции, так и на машины повышенной мощности. Поэтому при их написании надо пользоваться стандартными подмножествами языков Fortran и C, поскольку установка программ на удалённых машинах предполагает повторную трансляцию этих программ.

3. Программы типа оболочки, служащие для организации цепочек обращений к программам типа 1 и 2. Эти программы должны быть максимально лабильными, интерфейс должен быть максимально гибким. Это могут быть программы-однодневки, созданные под определённое исследование и под конкретного исследователя. Для таких программ подходящей системой программирования является Delphi.

### **О документации для разработчиков**

В современном программостроении принято создавать подробную документацию для разработчиков, чтобы данный комплекс мог жить и после нас. Принятые системы разработки комплекса значительно облегчают эту трудную задачу. Современные GUIDE в версии 6.x MatLab и Fortran позволяют проще комментировать и структурировать программы, чем предыдущие системы программирования.

Конечно, языки MatLab и Fortran не равноценны по компактности представления вычислительных алгоритмов. MatLab является специализированным языком вычислений с матрицами и массивами. Поэтому тексты вычислительных программ на языке MatLab очень легко читаются даже без подробных комментариев. Fortran-90 содержит очень ограниченный набор встроенных матричных операторов. Поэтому программы получаются длинными и трудно читаемыми. Это обстоятельство побудило нас снабдить комплекс программ LevInfinite двумя наборами вычислительных программ, написав эти два набора одновременно на языках MatLab и Fortran. Можно считать, что вычислительная программа на языке MatLab является подробнейшим комментарием к тексту этой же программы на языке Fortran. Кроме того, Fortran-программы формата .exe работают значительно быстрее, чем функции в среде MatLab. Поэтому для особо крупных молекулярных моделей

вычислительные процедуры имеет смысл выполнять с помощью отдельно стоящих exe-программ.

Что же касается форматов исходных данных для обмена между программами, то это было самое узкое место в наших предыдущих комплексах программ. Разработчикам нужно было внимательно читать тексты программ, чтобы узнать форматы данных. В данном комплексе программ весь обмен информацией ведётся через текстовые файлы, которые можно просто прочесть глазами. Вот и форматы данных - они перед тобой.

Так что есть шанс оставить будущим разработчикам вполне читаемую самодокументированную систему программ.