Brief Prefix	Explicit Concept	Summarizing Names of Evolvix Prefix Building Blocks Draft summary of how to use the Concepts [Table version QQ0r4]	StayVS version	Line
		Item Prefix Punctuation Sequences		1
.i	Item i	Items are for modeling what exists; Item Name i and its Synonyms all link to 1 main Feature explained by its Code in the Item's main Bag; to ZoomIn is to call .AnySynonym\i to produce Content from Code formally '$&'customized and ~randomized by the compiler if allowed	RR1	2
:i	Type i	Types are for modeling what *may* exist; :i is a Set of Rules describing any generalized or specialized Concept as abstract model; an Item is of AnyType, its own ItemProtoType, and of UnknownType until the compiler proves to which KnownType an ItemType belongs; it is good for finding errors *if* Types map reality well (the hardest work in Evolvix)	RR1	3
@i	View i	Views are for modeling what is measurable within a Boundary (Here), at it (Near), beyond it (There), or by shortcut (Fare): overview @i exists to ZoomOut to the Dictionary @i organizing Context @i with a nested Feature Population of Content Items and all contextual meta Items (for modeling emerging traits of the system or simply for observing it)	QQ1	4
.i\.x	Name.x	Define .x as perfect Synonym for .i; use \:MyDialect for pretty printing	QQ1	5
~i	Dice i	Uncertainty ~i affects .i; '$&' may auto switch precision and DiceType	QQ1	6
		Connection Prefix Punctuation Sequences		7
_i	Link i	Link _i can be Responsible, Fragile, etc as managed by .i (and '|$&%')	QQ1	8
|i	Pipe i	Filter Flow of Data over _i in .i to enforce all Rules in |i (eg firewalls etc)	QQ1	9
..i	Role i	Role of Input or Output ..i as Context defines it; easy to memoize	QQ1	10
?i	Asks i	Query ?i digs for extra output, assumptions, tests, logics, debugs, etc	QQ1	11
		Individualization Prefix Punctuation Sequences		12
$i	Uses i	Plan $i knows resources, implementations, tracks costs, credits, etc	PP1	13
&i	Flag i	Pick &i to switch assumptions, goals, methods, modes, options, etc on how to use or do .i (if not AsUsual); must all be predefined (like $)	PP1	14
%i	Spec i	Customize Machine x %Setup, %%Configuration, %%%InnerHacks	PP1	15
		Documentation Prefix Punctuation Sequences		16
#i	Tag i	Mark findings to aid search, evaluate in Batch of #: i, ToDo, Bug, ….	QQ1	17
!i	Info i	Human Info *only*; sort by Types i,doc,…. to compile literate programs	SS1	18
		Structural Prefix Punctuation Sequences (may all exist in Item i)		19
{}	Wall	Closed local Dictionary Node that maps all local Synonym Names to their relevant Explanations for this Context; no auto export or import	QQ1	20
()	Well	Open local Content Form: auto import, export of locally visible names	PP1	21
[]	List	Set builder notation describing Elements of Sets in Sets with diverse, dense, sparse, etc Dimensions; morphs into Table, Array, Alignment, Net, Database, Package, Filesystem etc via Evolvix VM Transformer	QQ1	22
'yz'	Pile	Unordered Bag of characters to choose from (eg 1 of 'yz' gets y or z)	QQ1	23
"yz"	Text	Sequence of characters (Types control UniCode normalizations, etc)	RR1	24
,	Chop	Separate next from previous elements in a List, Array, Table, [ ] etc	RR1	25
;	Stop	Separate next from previous sections in code; reset parsing; parse in parallel; structure compiled Info docs; pause compute to cooperate	RR1	26
`	Glue	Build Prefix Sequences via genitive relations (Car`Key = Key of a Car)	QQ1	27
[1] i	Word i	Keyword or Prefix must follow [1]Whitespace (if not in a special Context)	RR1	28

Figure P0: Core concepts for Building Prefix Punctuation Sequences in Evolvix. Languages aiming for a stable extensible user-friendly vocabulary have to **(i)** maximize space for growing lists of keywords and **(ii)** keep it free from confusing ones. Thus, Evolvix **(i)** reserves all words as keywords by expecting all names from users to get a useful prefix (lines 2 to 17); **(ii)** Evolvix tracks formal milestones on the journey to stability for simplifying fast removal of avoidable complexity (with StayVS, a stabilizing versioning system in the works for assessing stabilities of Elements and Sets independently; see Fig.REF). To mark likely Evolvix Keywords, we capitalize Concepts in Core Evolvix (like 'Bag' for 'container of varying size') and drop dots from useful Latin shortcuts (like eg for e.g.: or etc for etc.).