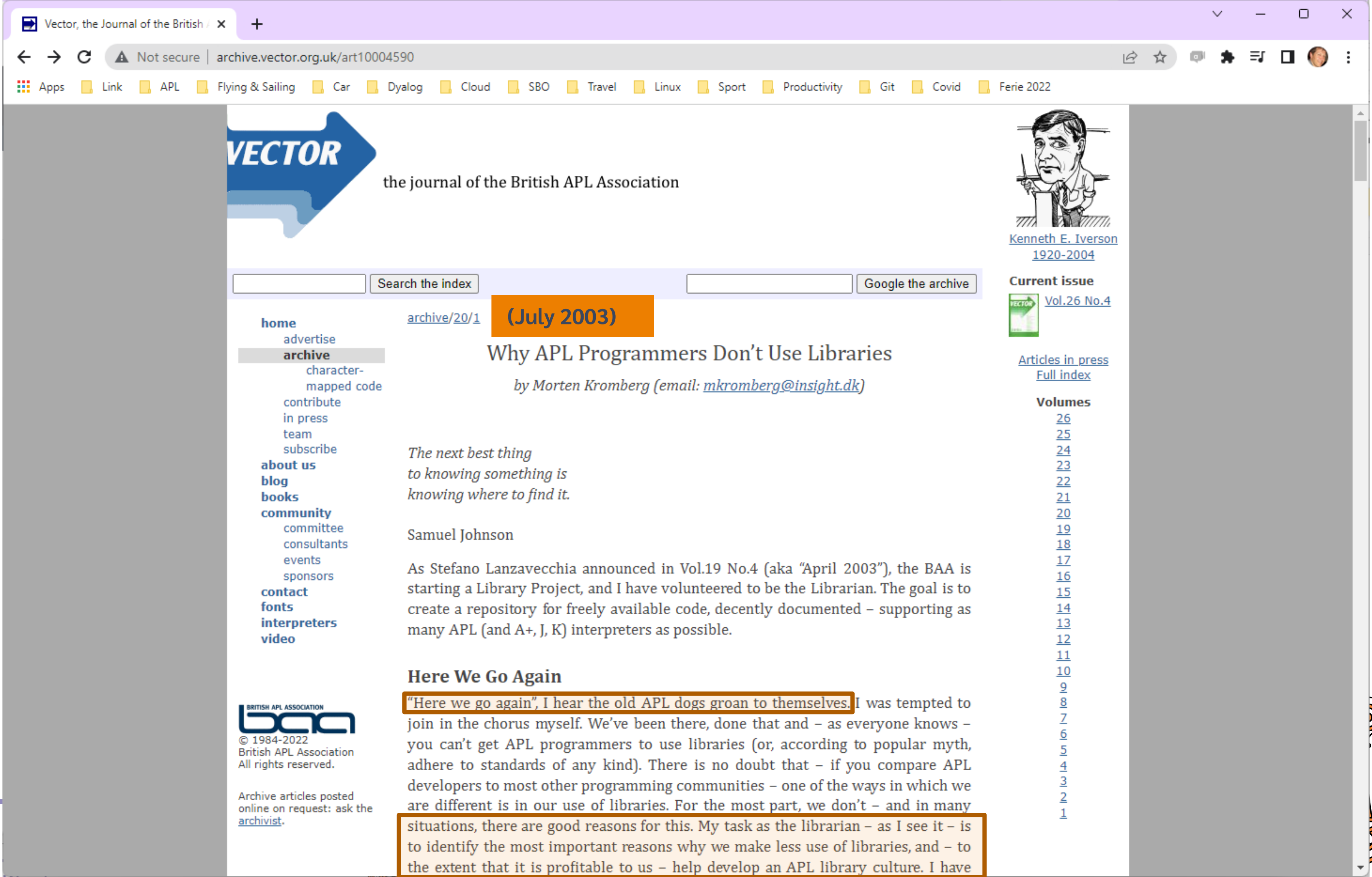
Olhão 2022

The "P" Words
Projects and **P**ackages

Morten Kromberg

Agenda

- Packages
  - Why do we need packages?
  - Getting started with Tatin
  - Using a Tatin Package
- Projects
  - Why do we need project managers?
  - Acre & Dado
  - Cider



the journal of the British APL Association

archive/20/1

(July 2003)

Why APL Programmers Don't Use Libraries

by Morten Kromberg (email: mkromberg@insight.dk)

The next best thing
to knowing something is
knowing where to find it.

Samuel Johnson

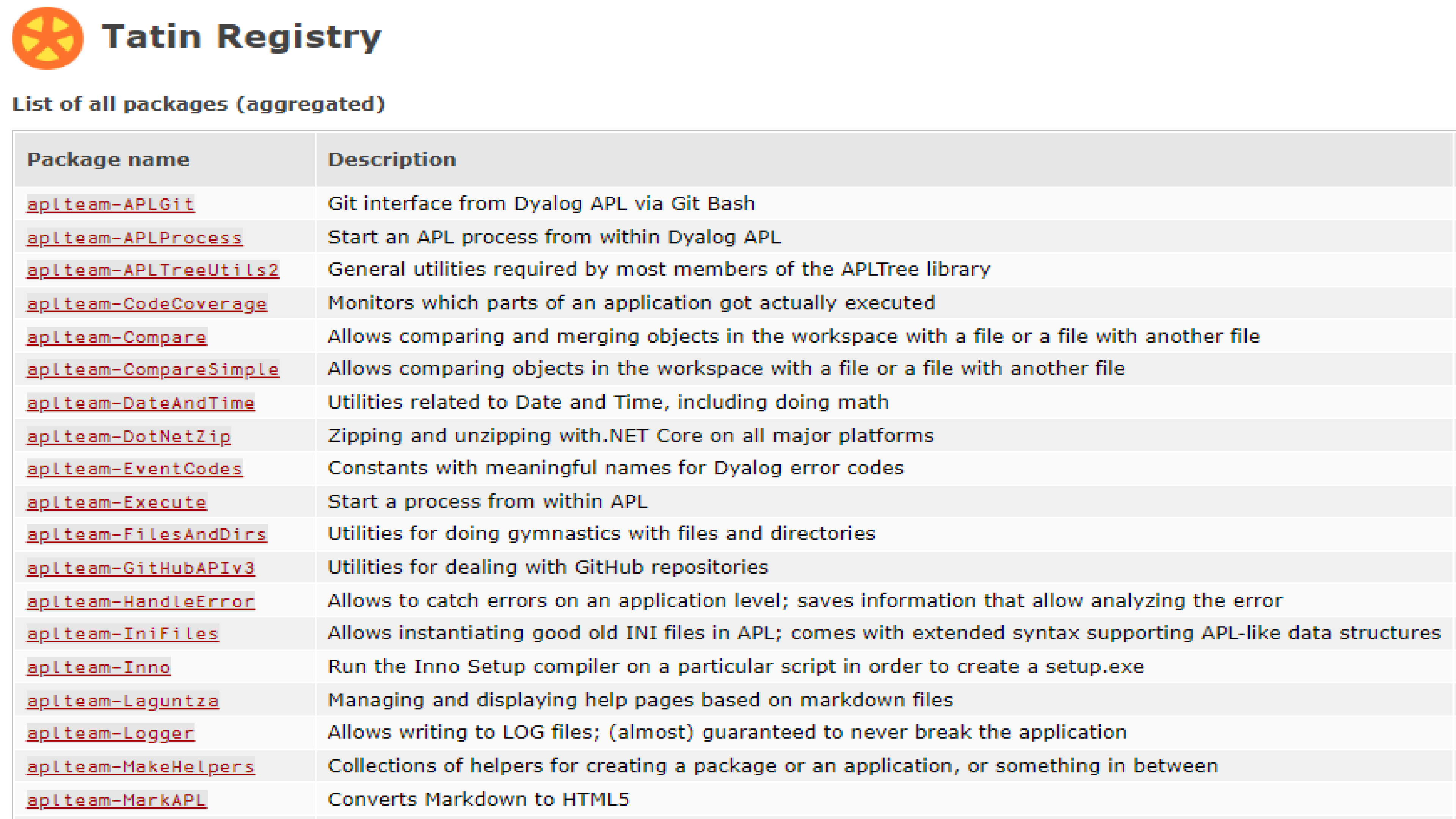
As Stefano Lanzavecchia announced in Vol.19 No.4 (aka "April 2003"), the BAA is starting a Library Project, and I have volunteered to be the Librarian. The goal is to create a repository for freely available code, decently documented – supporting as many APL (and A+, J, K) interpreters as possible.

Here We Go Again

"Here we go again", I hear the old APL dogs groan to themselves. I was tempted to join in the chorus myself. We've been there, done that and – as everyone knows – you can't get APL programmers to use libraries (or, according to popular myth, adhere to standards of any kind). There is no doubt that – if you compare APL developers to most other programming communities – one of the ways in which we are different is in our use of libraries. For the most part, we don't – and in many situations, there are good reasons for this. My task as the librarian – as I see it – is to identify the most important reasons why we make less use of libraries, and – to the extent that it is profitable to us – help develop an APL library culture. I have

Most Importantly

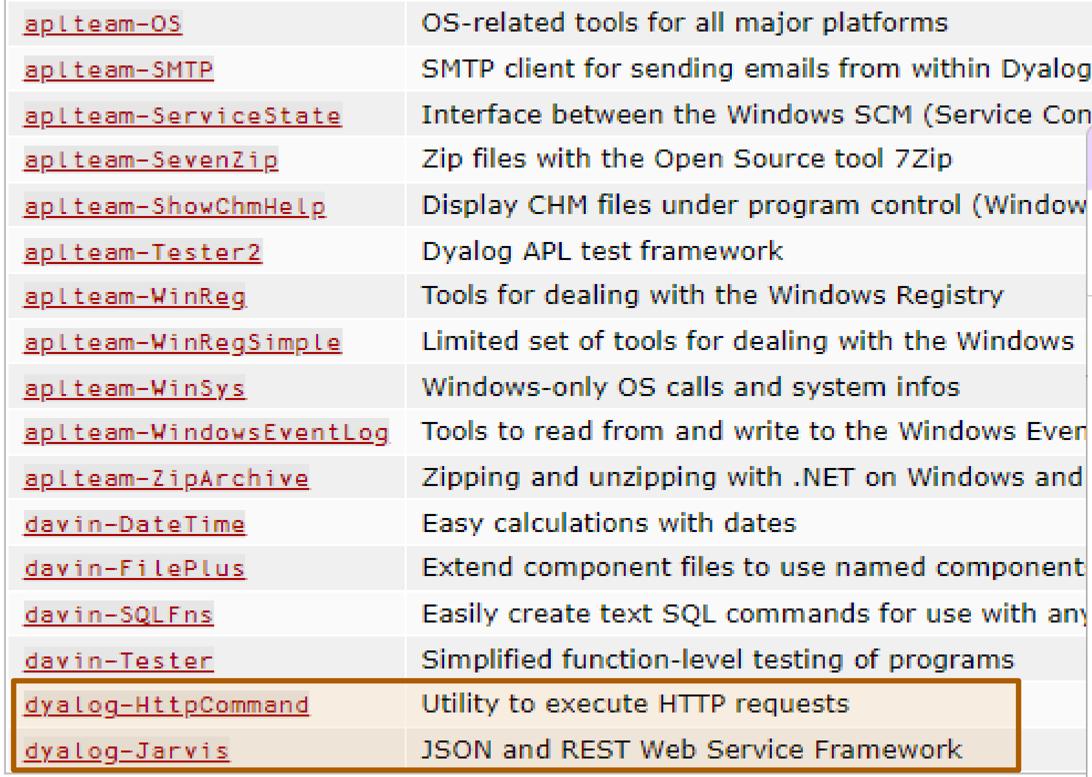
- In addition to be valuable educational resources…
- Packages are critical to keeping APL competitive as a tool for building modern applications
- Support web protocols and components, data formats, operating system APIs, security requirements



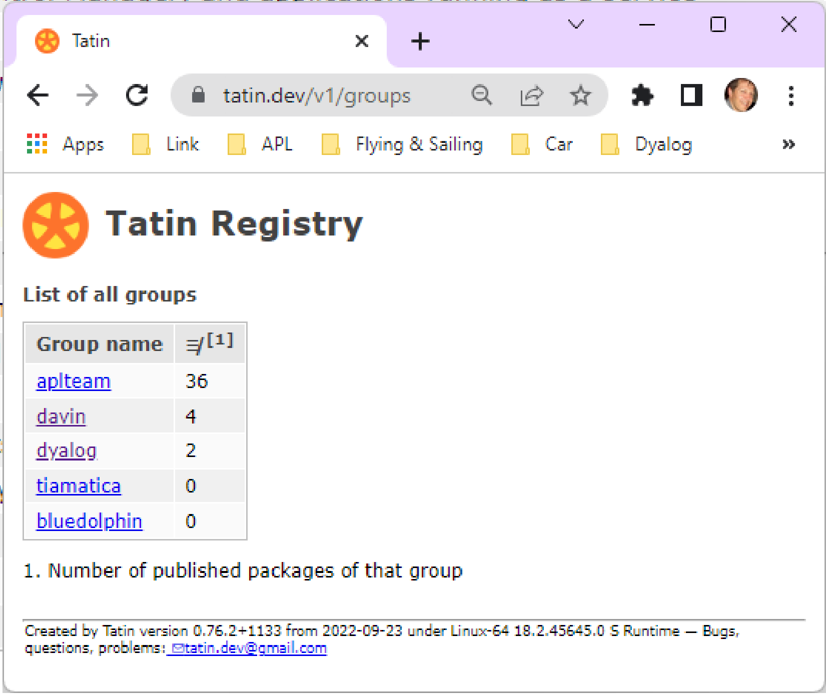
Tatin Registry

List of all packages (aggregated)

Package name	Description
aplteam-APLGit	Git interface from Dyalog APL via Git Bash
aplteam-APLProcess	Start an APL process from within Dyalog APL
aplteam-APLTreeUtils2	General utilities required by most members of the APLTree library
aplteam-CodeCoverage	Monitors which parts of an application got actually executed
aplteam-Compare	Allows comparing and merging objects in the workspace with a file or a file with another file
aplteam-CompareSimple	Allows comparing objects in the workspace with a file or a file with another file
aplteam-DateAndTime	Utilities related to Date and Time, including doing math
aplteam-DotNetZip	Zipping and unzipping with.NET Core on all major platforms
aplteam-EventCodes	Constants with meaningful names for Dyalog error codes
aplteam-Execute	Start a process from within APL
aplteam-FilesAndDirs	Utilities for doing gymnastics with files and directories
aplteam-GitHubAPIv3	Utilities for dealing with GitHub repositories
aplteam-HandleError	Allows to catch errors on an application level; saves information that allow analyzing the error
aplteam-IniFiles	Allows instantiating good old INI files in APL; comes with extended syntax supporting APL-like data structures
aplteam-Inno	Run the Inno Setup compiler on a particular script in order to create a setup.exe
aplteam-Laguntza	Managing and displaying help pages based on markdown files
aplteam-Logger	Allows writing to LOG files; (almost) guaranteed to never break the application
aplteam-MakeHelpers	Collections of helpers for creating a package or an application, or something in between
aplteam-MarkAPL	Converts Markdown to HTML5



aplteam-OS	OS-related tools for all major platforms
aplteam-SMTP	SMTP client for sending emails from within Dyalog APL
aplteam-ServiceState	Interface between the Windows SCM (Service Control Manager) and applications running as a service
aplteam-SevenZip	Zip files with the Open Source tool 7Zip
aplteam-ShowChmHelp	Display CHM files under program control (Window
aplteam-Tester2	Dyalog APL test framework
aplteam-WinReg	Tools for dealing with the Windows Registry
aplteam-WinRegSimple	Limited set of tools for dealing with the Windows
aplteam-WinSys	Windows-only OS calls and system infos
aplteam-WindowsEventLog	Tools to read from and write to the Windows Even
aplteam-ZipArchive	Zipping and unzipping with .NET on Windows and
davin-DateTime	Easy calculations with dates
davin-FilePlus	Extend component files to use named component
davin-SQLFns	Easily create text SQL commands for use with any
davin-Tester	Simplified function-level testing of programs
dyalog-HttpCommand	Utility to execute HTTP requests
dyalog-Jarvis	JSON and REST Web Service Framework



Tatin Registry

List of all groups

Group name	≢[1]
aplteam	36
davin	4
dyalog	2
tiamatica	0
bluedolphin	0

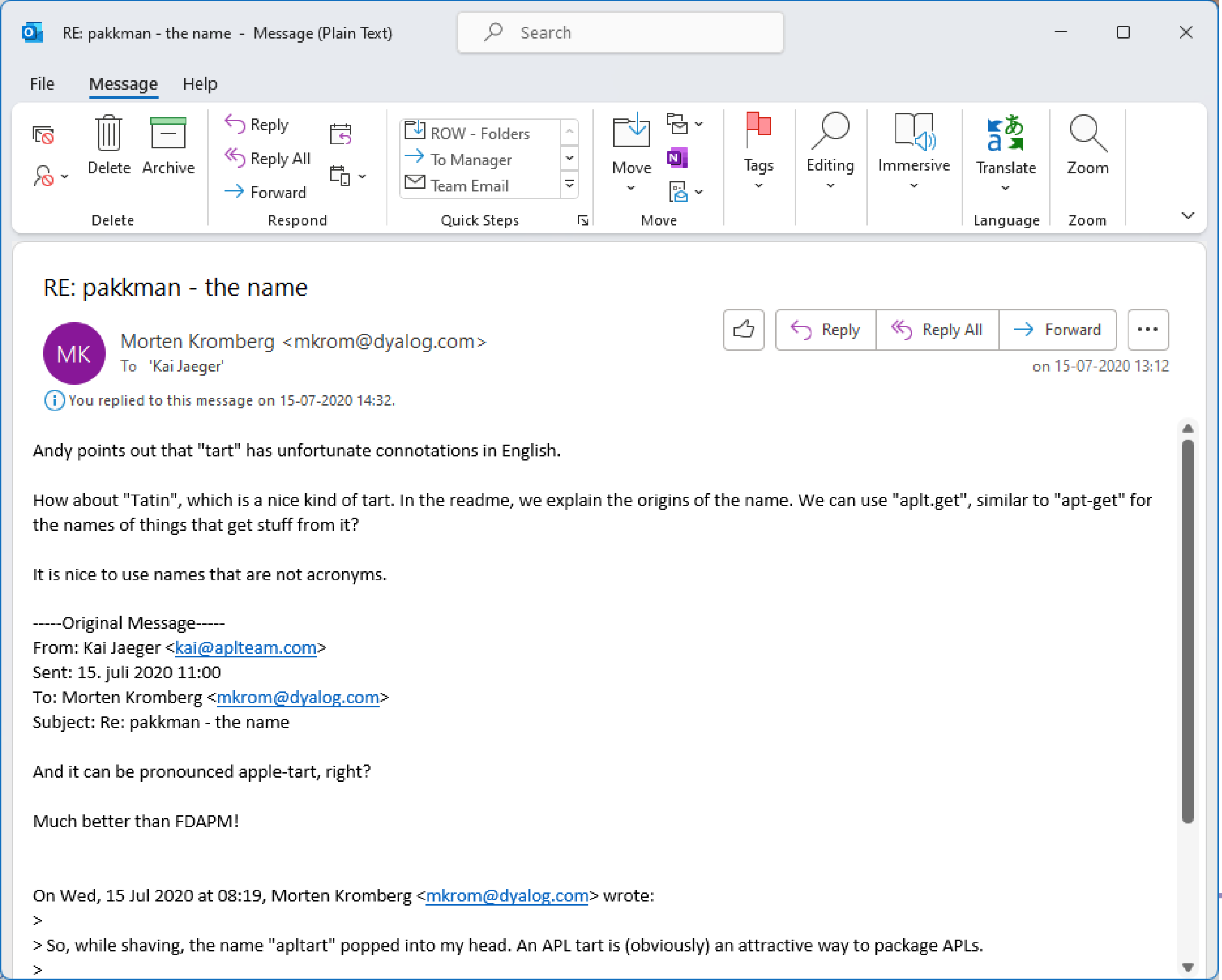
1. Number of published packages of that group

Created by Tatin version 0.76.2+1133 from 2022-09-23 under Linux-64 18.2.45645.0 S Runtime — Bugs, questions, problems: tatin.dev@gmail.com

Finally, the first Dyalog Packages are published!

Introducing Tatin

- Developed by Kai Jäger (first lines of code written in 2020)
- Funded by Dyalog
- Designed by Kai and Gilgamesh Athoraya
- Input from various people at Dyalog
- Logo by Adam Brudzewsky
- Many thanks to Davin Church, the first real user of the system other than Kai himself
- Paul Mansour is not to blame for the current design of Tatin, but has been an important inspiration



RE: pakkman - the name

MK Morten Kromberg <mkrom@dyalog.com>
To 'Kai Jaeger'
on 15-07-2020 13:12

You replied to this message on 15-07-2020 14:32.

Andy points out that "tart" has unfortunate connotations in English.

How about "Tatin", which is a nice kind of tart. In the readme, we explain the origins of the name. We can use "aplt.get", similar to "apt-get" for the names of things that get stuff from it?

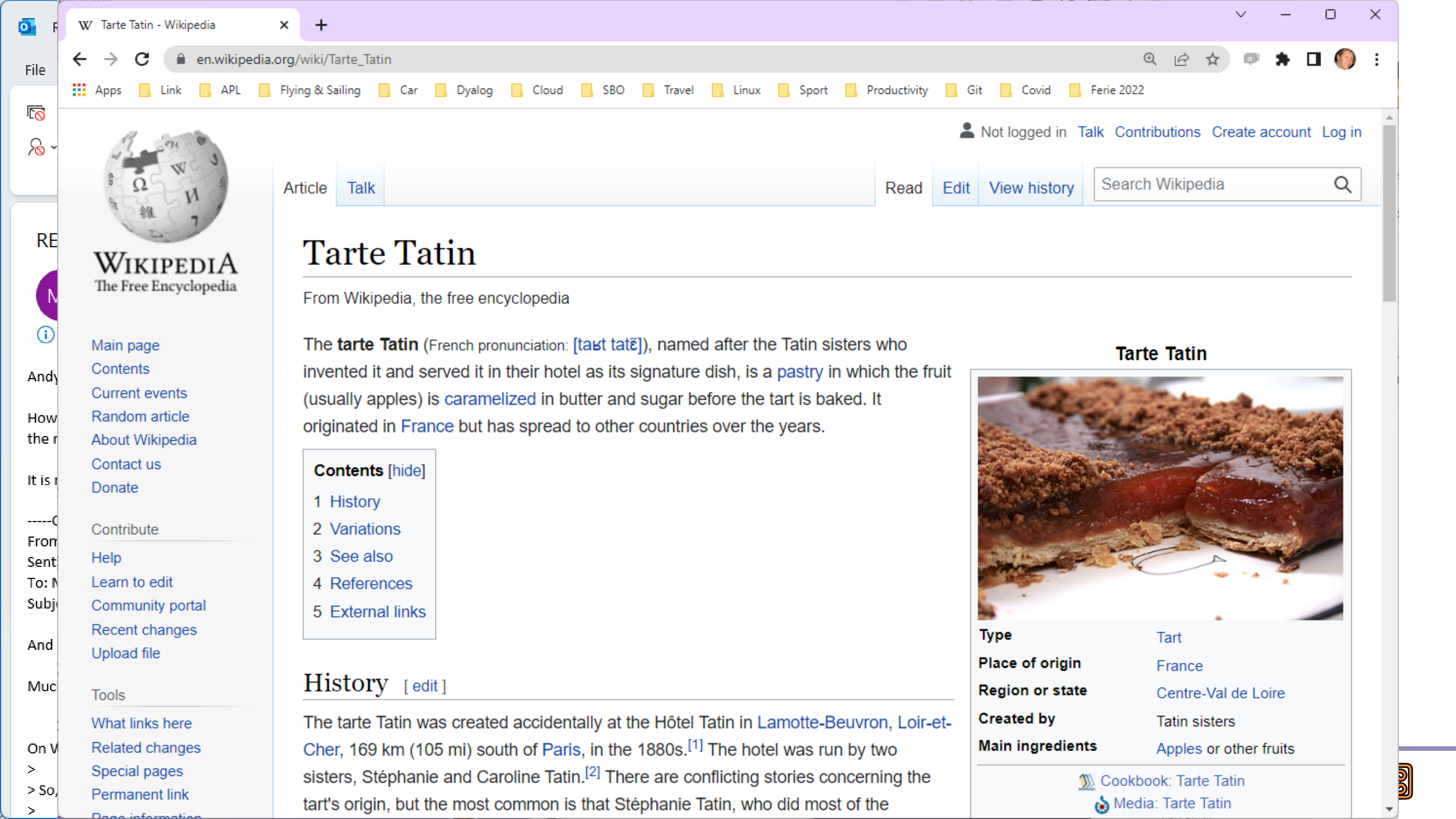
It is nice to use names that are not acronyms.

-----Original Message-----
From: Kai Jaeger <kai@aplteam.com>
Sent: 15. juli 2020 11:00
To: Morten Kromberg <mkrom@dyalog.com>
Subject: Re: pakkman - the name

And it can be pronounced apple-tart, right?

Much better than FDAPM!

On Wed, 15 Jul 2020 at 08:19, Morten Kromberg <mkrom@dyalog.com> wrote:
>
> So, while shaving, the name "apltart" popped into my head. An APL tart is (obviously) an attractive way to package APLs.
>



Tarte Tatin

From Wikipedia, the free encyclopedia

The **tarte Tatin** (French pronunciation: [taʁt tatɛ̃]), named after the Tatin sisters who invented it and served it in their hotel as its signature dish, is a pastry in which the fruit (usually apples) is caramelized in butter and sugar before the tart is baked. It originated in France but has spread to other countries over the years.

Tarte Tatin

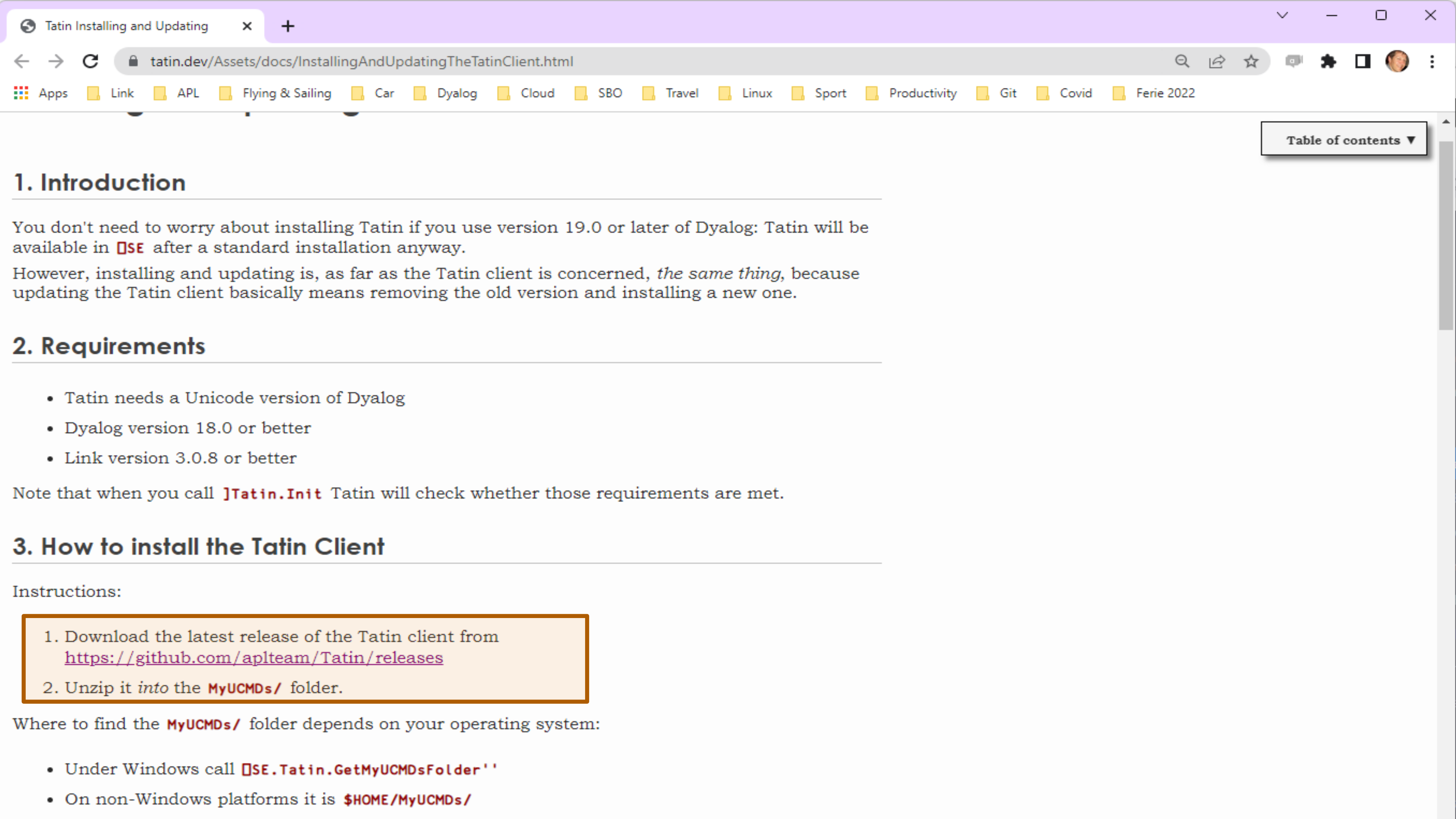
Type	Tart
Place of origin	France
Region or state	Centre-Val de Loire
Created by	Tatin sisters
Main ingredients	Apples or other fruits

Cookbook: Tarte Tatin
Media: Tarte Tatin

Contents [hide]

History [edit]

The tarte Tatin was created accidentally at the Hôtel Tatin in Lamotte-Beuvron, Loir-et-Cher, 169 km (105 mi) south of Paris, in the 1880s.[1] The hotel was run by two sisters, Stéphanie and Caroline Tatin.[2] There are conflicting stories concerning the tart's origin, but the most common is that Stéphanie Tatin, who did most of the



1. Introduction

You don't need to worry about installing Tatin if you use version 19.0 or later of Dyalog: Tatin will be available in `⎕SE` after a standard installation anyway.

However, installing and updating is, as far as the Tatin client is concerned, *the same thing*, because updating the Tatin client basically means removing the old version and installing a new one.

2. Requirements

- Tatin needs a Unicode version of Dyalog
- Dyalog version 18.0 or better
- Link version 3.0.8 or better

Note that when you call `]Tatin.Init` Tatin will check whether those requirements are met.

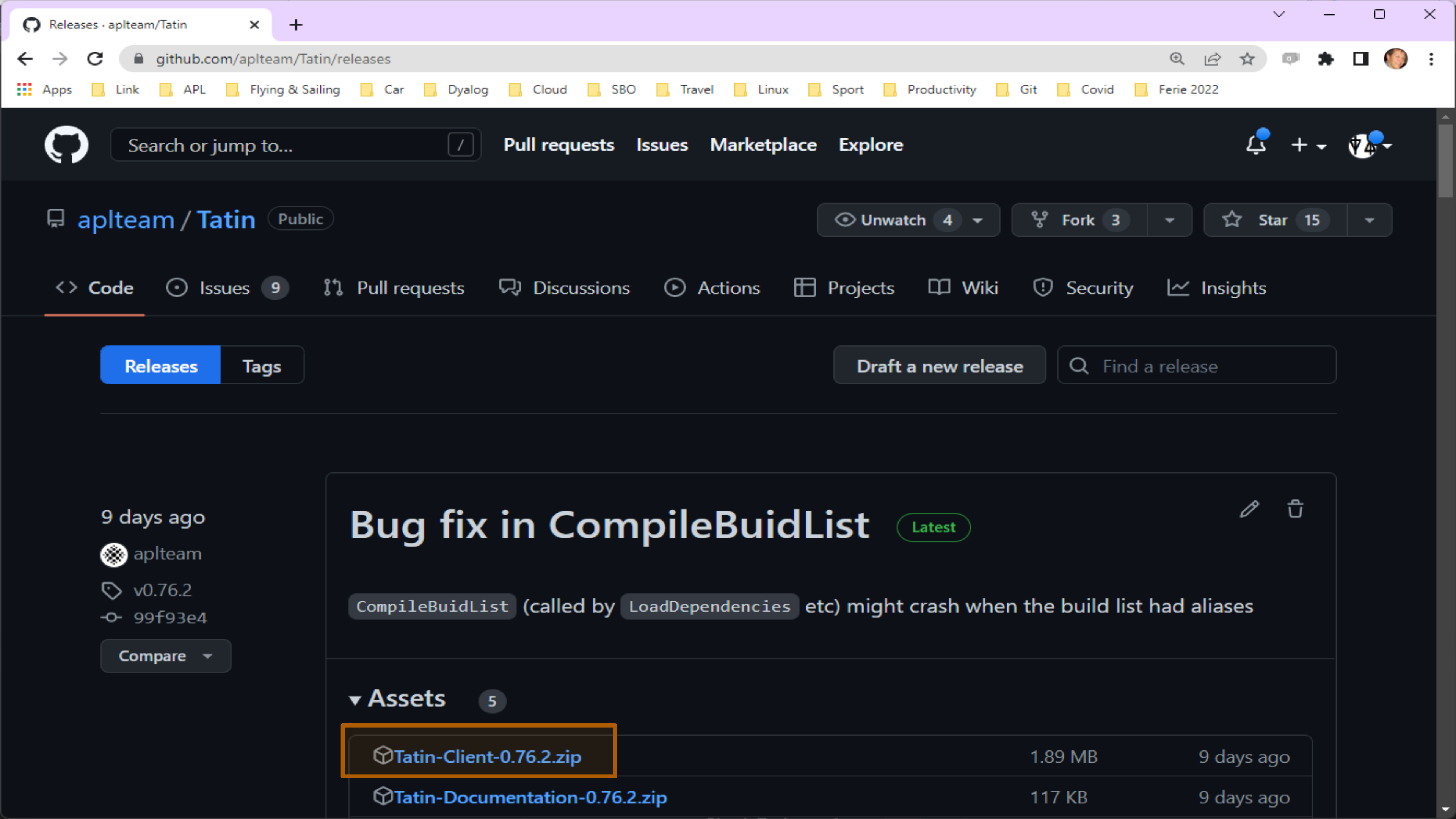
3. How to install the Tatin Client

Instructions:

1. Download the latest release of the Tatin client from https://github.com/aplteam/Tatin/releases
2. Unzip it *into* the `MyUCMDs/` folder.

Where to find the `MyUCMDs/` folder depends on your operating system:

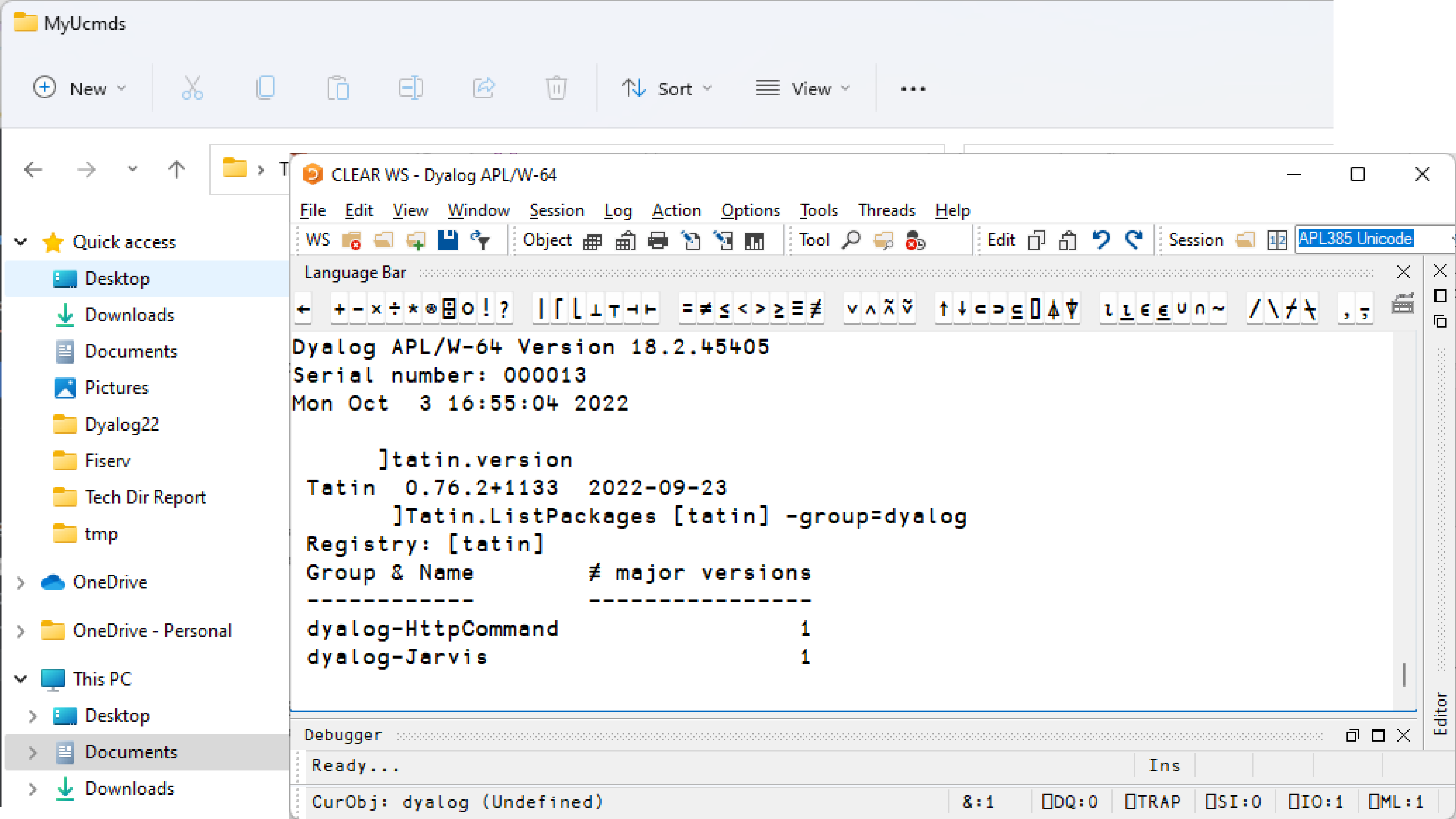
- Under Windows call `⎕SE.Tatin.GetMyUCMDsFolder''`
- On non-Windows platforms it is `$HOME/MyUCMDs/`



Bug fix in CompileBuidList

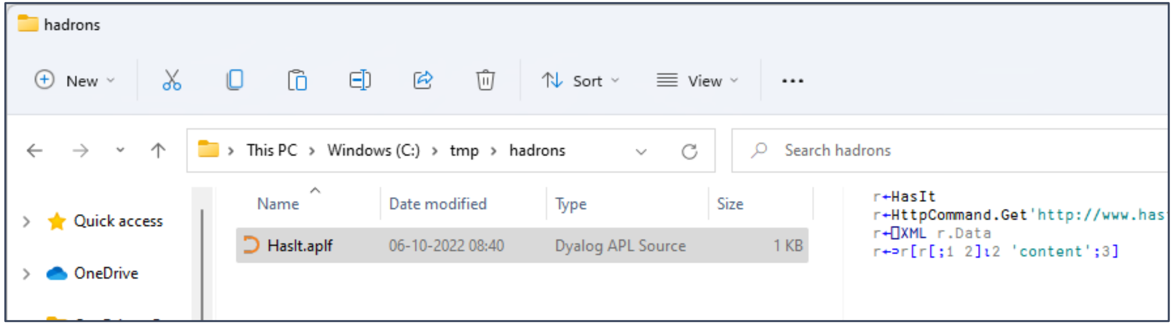
`CompileBuidList` (called by `LoadDependencies` etc) might crash when the build list had aliases

Tatin-Client-0.76.2.zip	1.89 MB	9 days ago
Tatin-Documentation-0.76.2.zip	117 KB	9 days ago



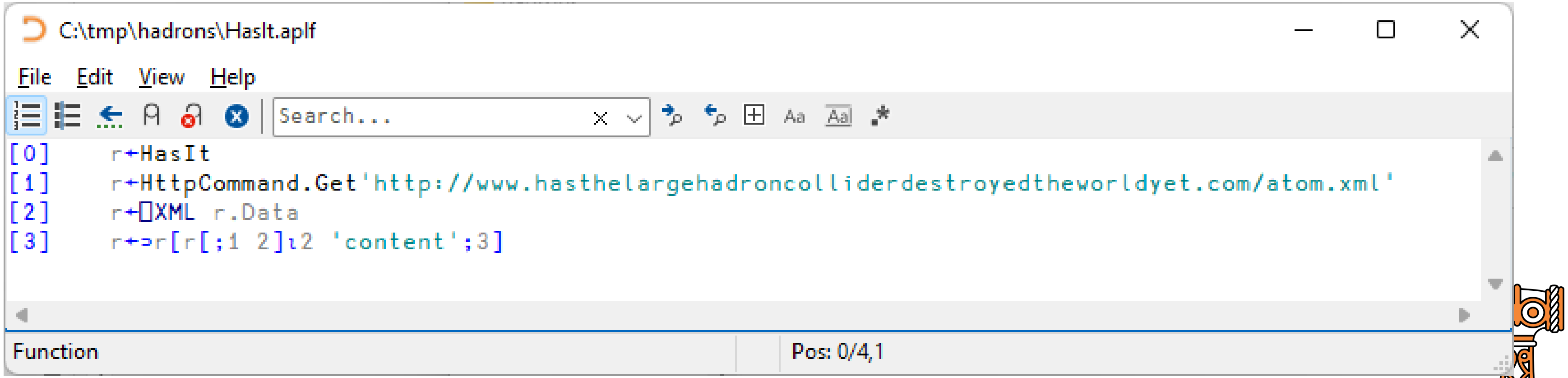
```
Dyalog APL/W-64 Version 18.2.45405
Serial number: 000013
Mon Oct  3 16:55:04 2022

      ]tatin.version
 Tatin  0.76.2+1133  2022-09-23
       ]Tatin.ListPackages [tatin] -group=dyalog
 Registry: [tatin]
 Group & Name        ≢ major versions
 ------------        ----------------
 dyalog-HttpCommand                 1
 dyalog-Jarvis                      1
```

A Small APL Application



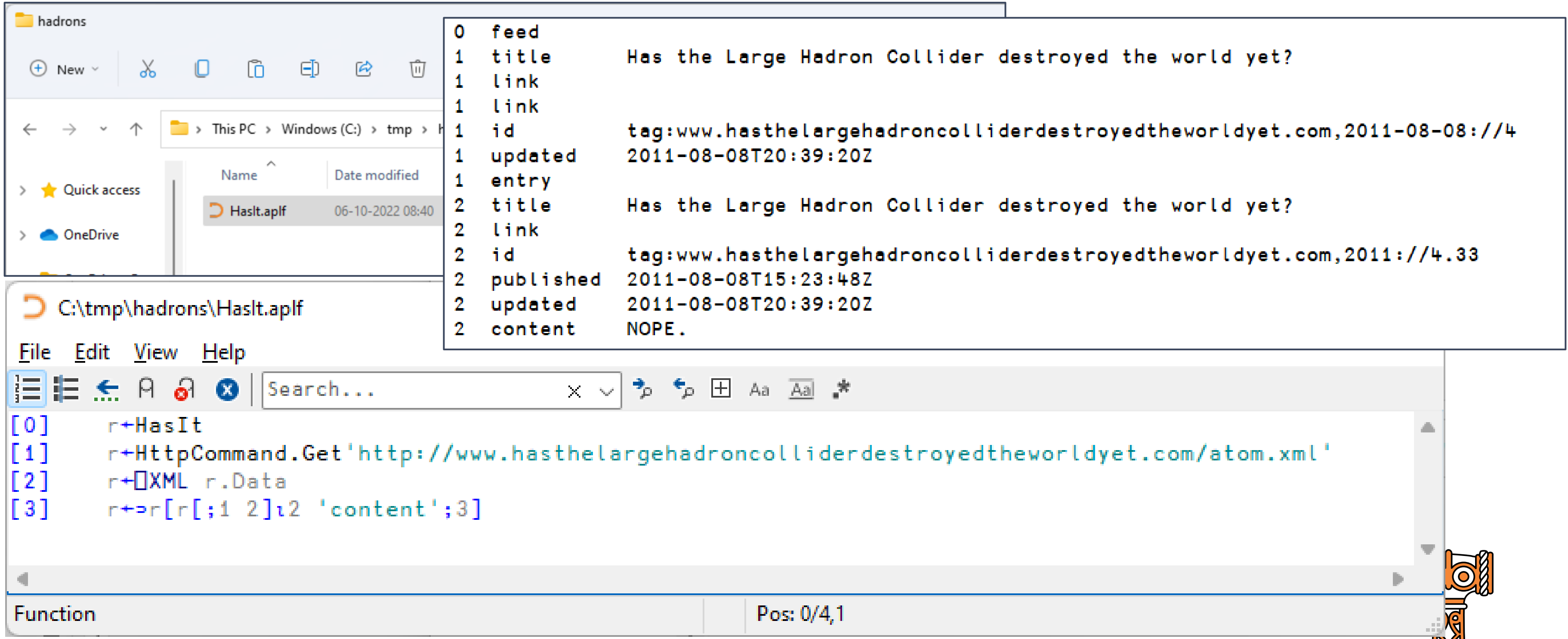
Name	Date modified	Type	Size
HasIt.aplf	06-10-2022 08:40	Dyalog APL Source	1 KB

```
r←HasIt
r←HttpCommand.Get'http://www.has
r←⎕XML r.Data
r←⊃r[r[;1 2]⍳2 'content';3]
```



```
r←HasIt
r←HttpCommand.Get'http://www.hasthelargehadroncolliderdestroyedtheworldyet.com/atom.xml'
r←⎕XML r.Data
r←⊃r[r[;1 2]⍳2 'content';3]
```

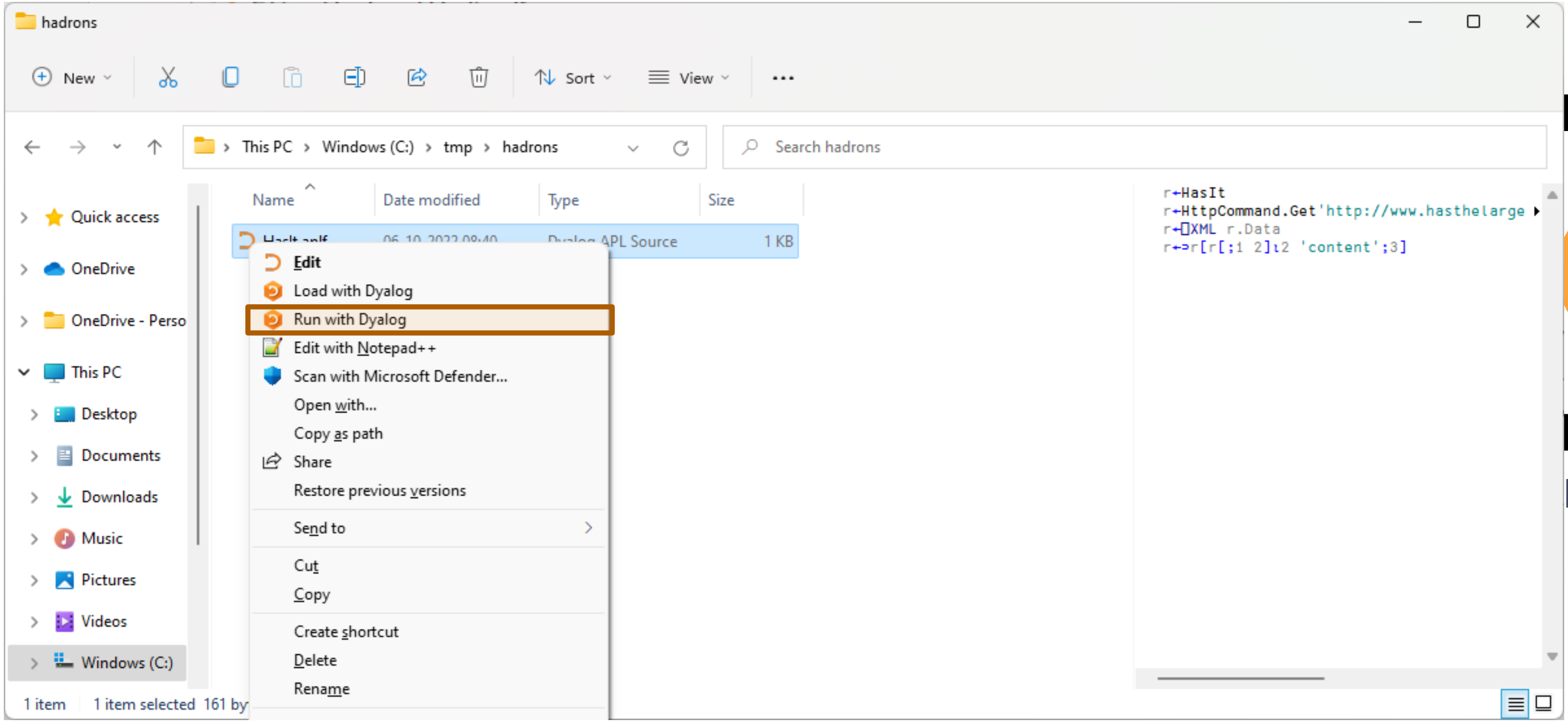
A Small APL Application



```
0  feed
1  title       Has the Large Hadron Collider destroyed the world yet?
1  link
1  link
1  id          tag:www.hasthelargehadroncolliderdestroyedtheworldyet.com,2011-08-08://4
1  updated     2011-08-08T20:39:20Z
1  entry
2  title       Has the Large Hadron Collider destroyed the world yet?
2  link
2  id          tag:www.hasthelargehadroncolliderdestroyedtheworldyet.com,2011://4.33
2  published   2011-08-08T15:23:48Z
2  updated     2011-08-08T20:39:20Z
2  content     NOPE.
```

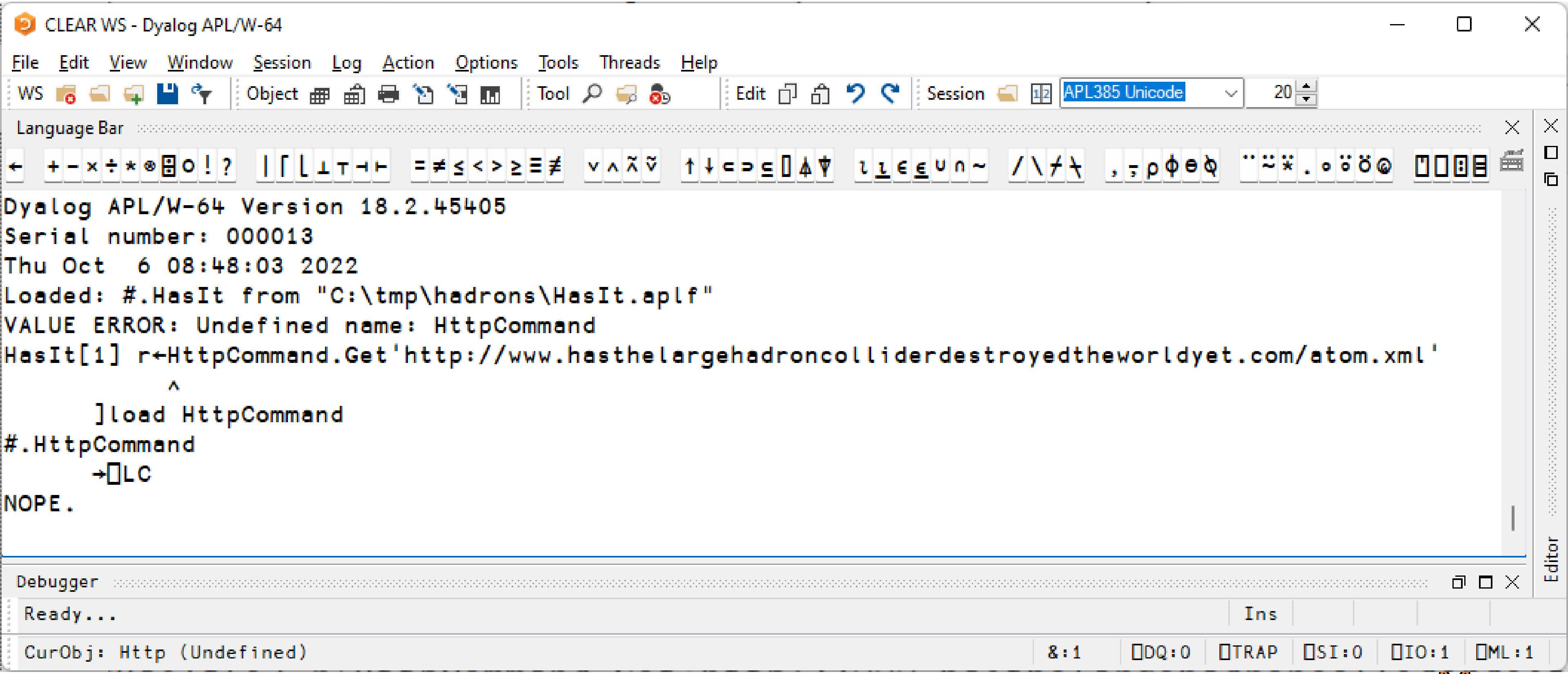
```
     r←HasIt
     r←HttpCommand.Get'http://www.hasthelargehadroncolliderdestroyedtheworldyet.com/atom.xml'
     r←⎕XML r.Data
     r←⊃r[r[;1 2]⍳2 'content';3]
```

A Small APL Application



```
r←HasIt
r←HttpCommand.Get'http://www.hasthelarge
r←⎕XML r.Data
r←⊃r[r[;1 2]⍳2 'content';3]
```

Link Logo

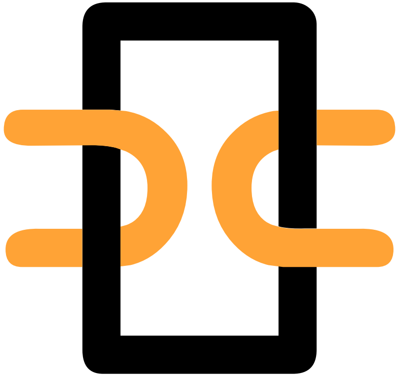

CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
Editor

```
Dyalog APL/W-64 Version 18.2.45405
Serial number: 000013
Thu Oct  6 08:48:03 2022
Loaded: #.HasIt from "C:\tmp\hadrons\HasIt.aplf"
VALUE ERROR: Undefined name: HttpCommand
HasIt[1] r←HttpCommand.Get'http://www.hasthelargehadroncolliderdestroyedtheworldyet.com/atom.xml'
           ∧
      ]load HttpCommand
#.HttpCommand
      →⎕LC
NOPE.
```

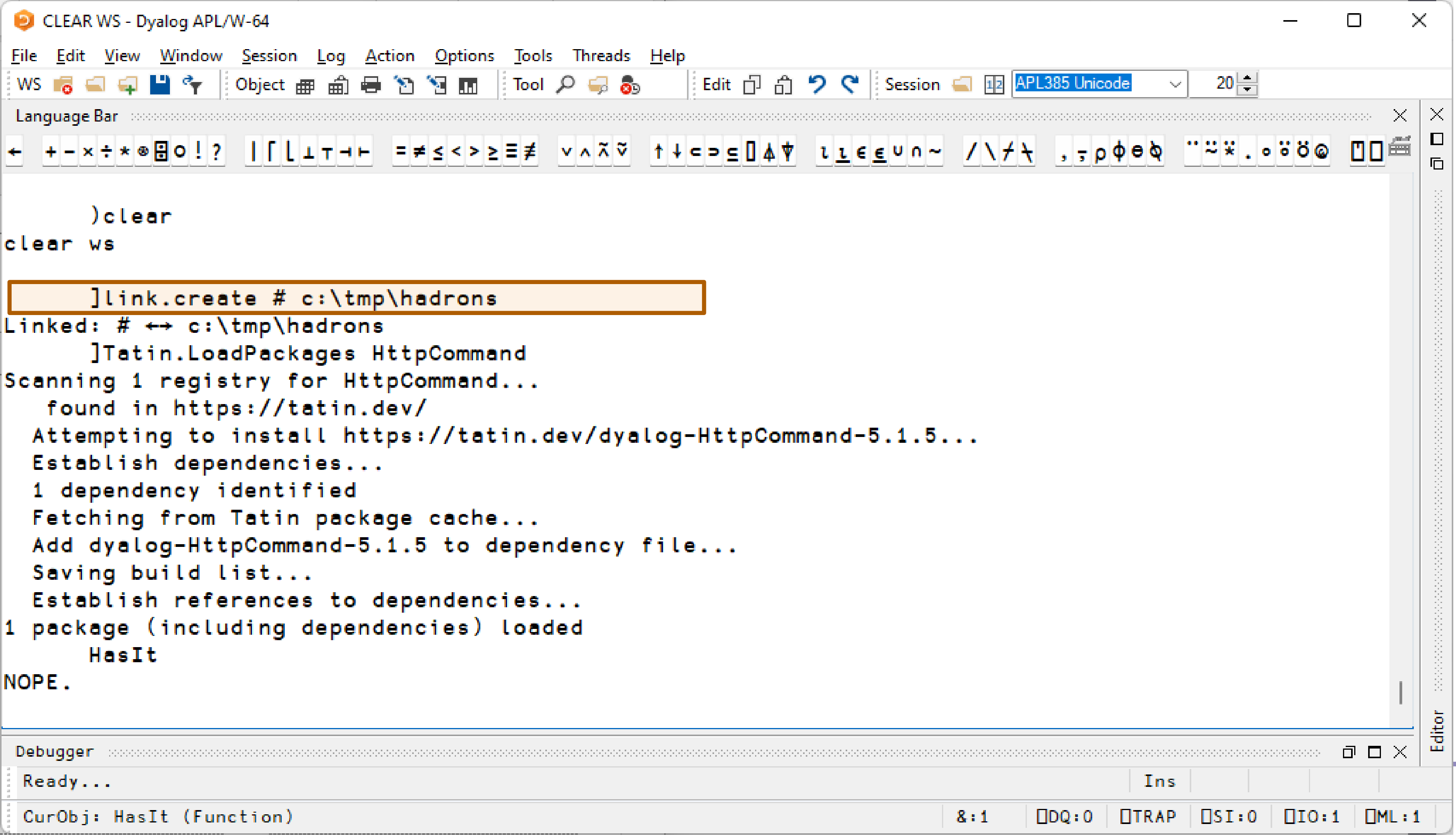
```
Ready...                                   Ins
CurObj: Http (Undefined)          &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```

A bit more structure

- Load our source with]Link.Create
- Load HttpCommand with]Tatin.LoadPackages

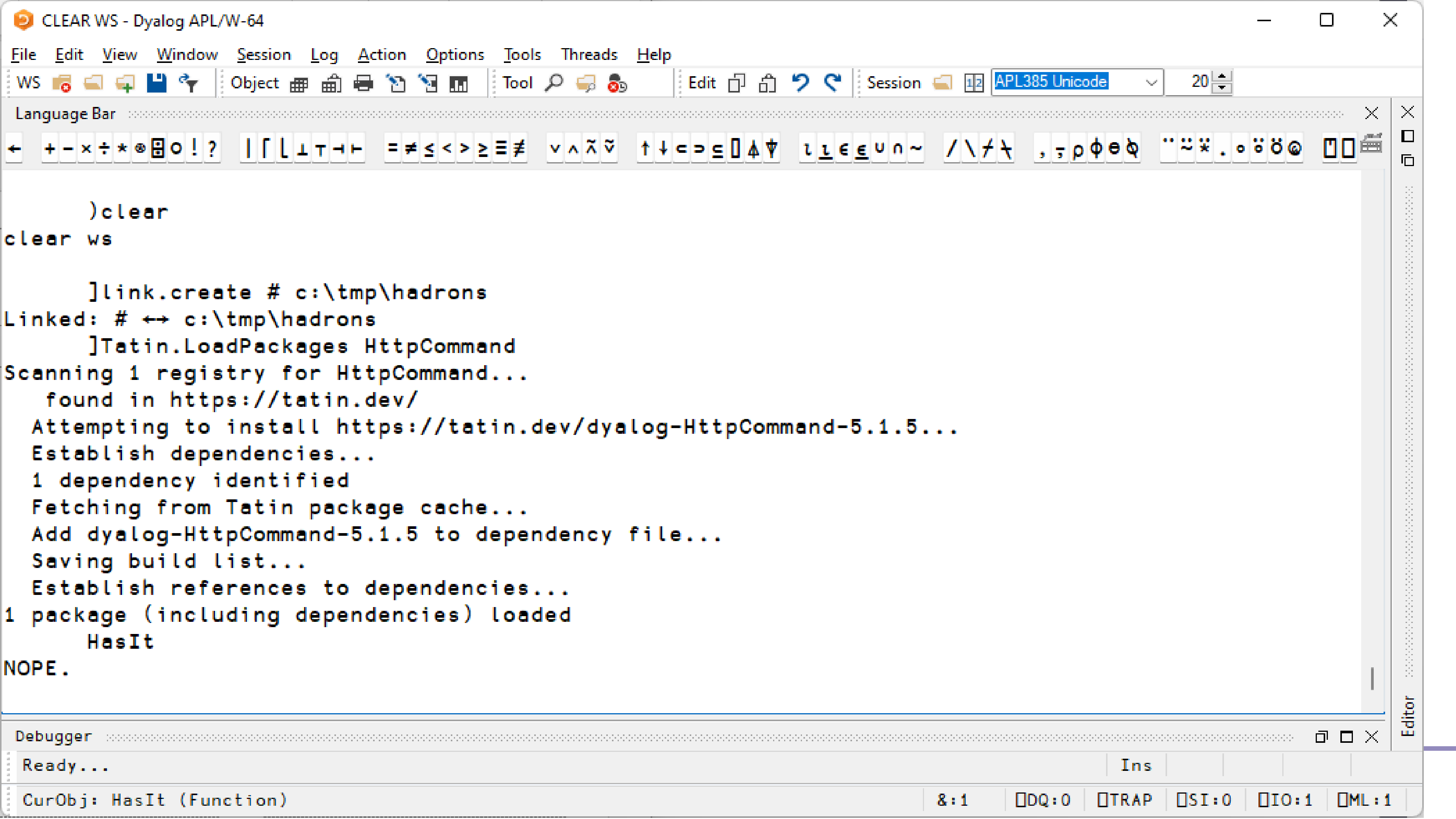


Proposed Link Logo


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
Editor

```
      )clear
clear ws

      ]link.create # c:\tmp\hadrons
Linked: # ↔ c:\tmp\hadrons
      ]Tatin.LoadPackages HttpCommand
Scanning 1 registry for HttpCommand...
   found in https://tatin.dev/
  Attempting to install https://tatin.dev/dyalog-HttpCommand-5.1.5...
  Establish dependencies...
  1 dependency identified
  Fetching from Tatin package cache...
  Add dyalog-HttpCommand-5.1.5 to dependency file...
  Saving build list...
  Establish references to dependencies...
1 package (including dependencies) loaded
      HasIt
NOPE.
```

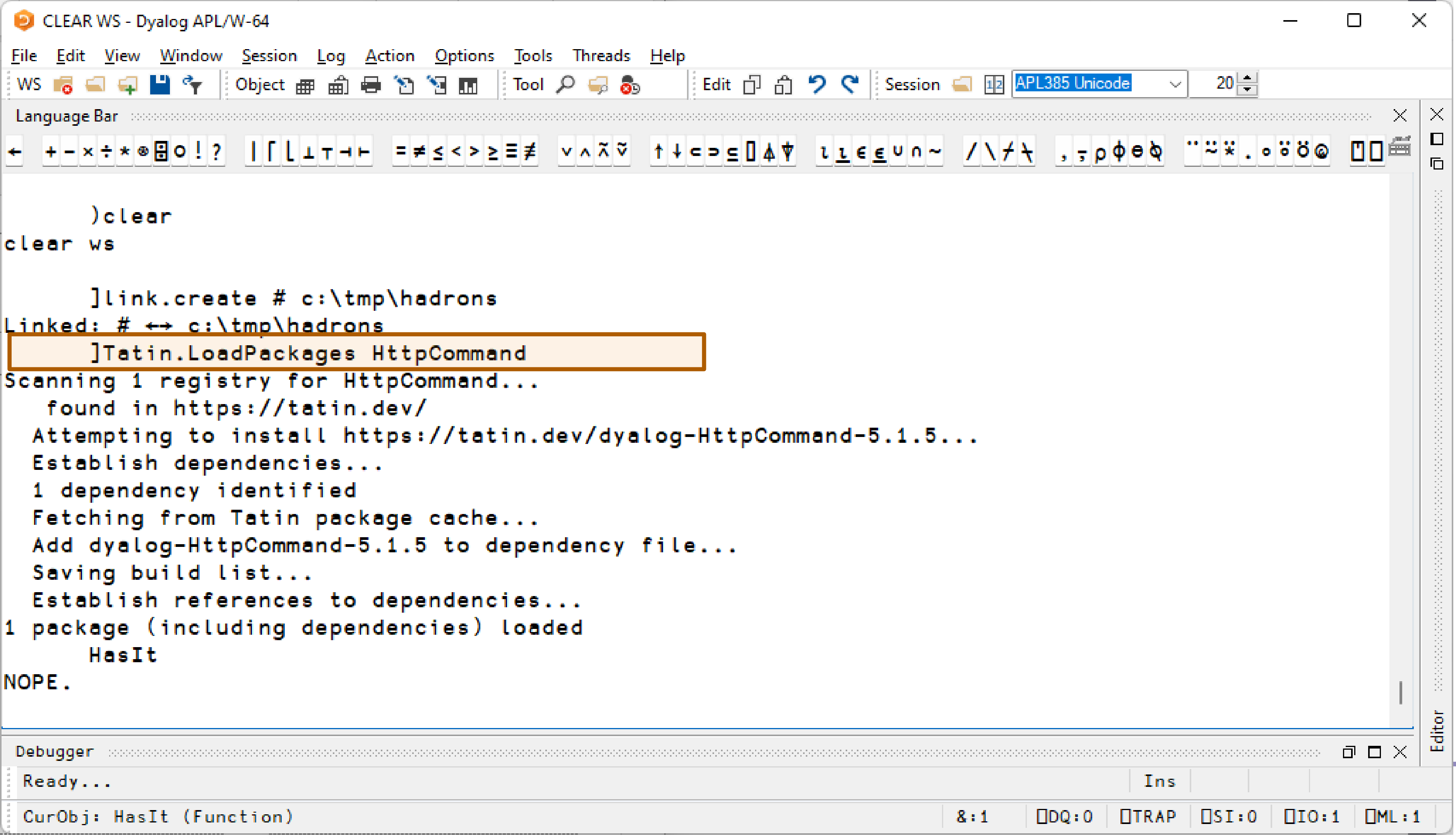
```
Ready...                                   Ins
CurObj: HasIt (Function)     &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```



CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
Editor

```
      )clear
clear ws

      ]link.create # c:\tmp\hadrons
Linked: # ←→ c:\tmp\hadrons
      ]Tatin.LoadPackages HttpCommand
Scanning 1 registry for HttpCommand...
   found in https://tatin.dev/
  Attempting to install https://tatin.dev/dyalog-HttpCommand-5.1.5...
  Establish dependencies...
  1 dependency identified
  Fetching from Tatin package cache...
  Add dyalog-HttpCommand-5.1.5 to dependency file...
  Saving build list...
  Establish references to dependencies...
1 package (including dependencies) loaded
      HasIt
NOPE.
```

```
Ready...                                    Ins
CurObj: HasIt (Function)        &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```

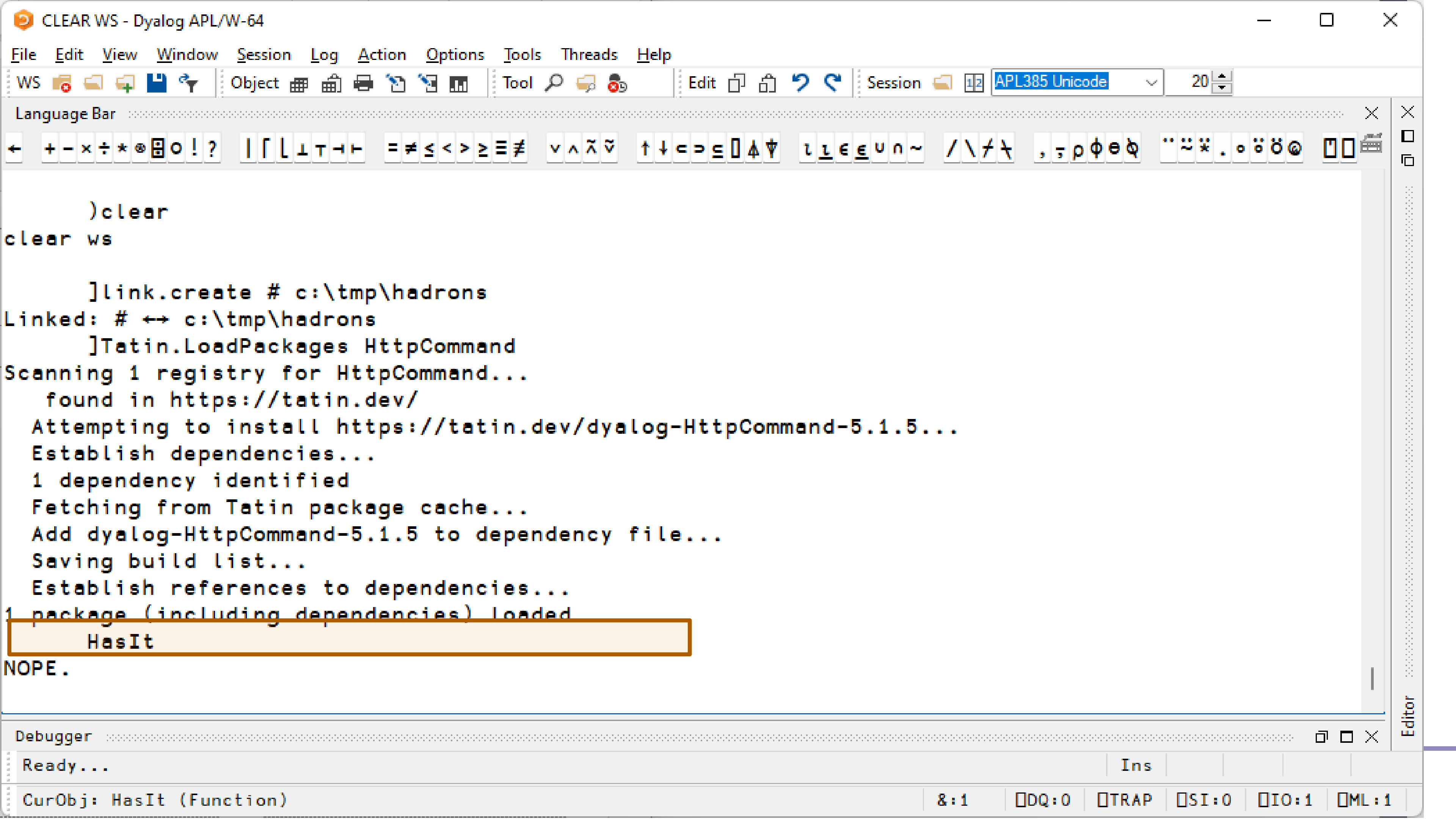


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
Editor

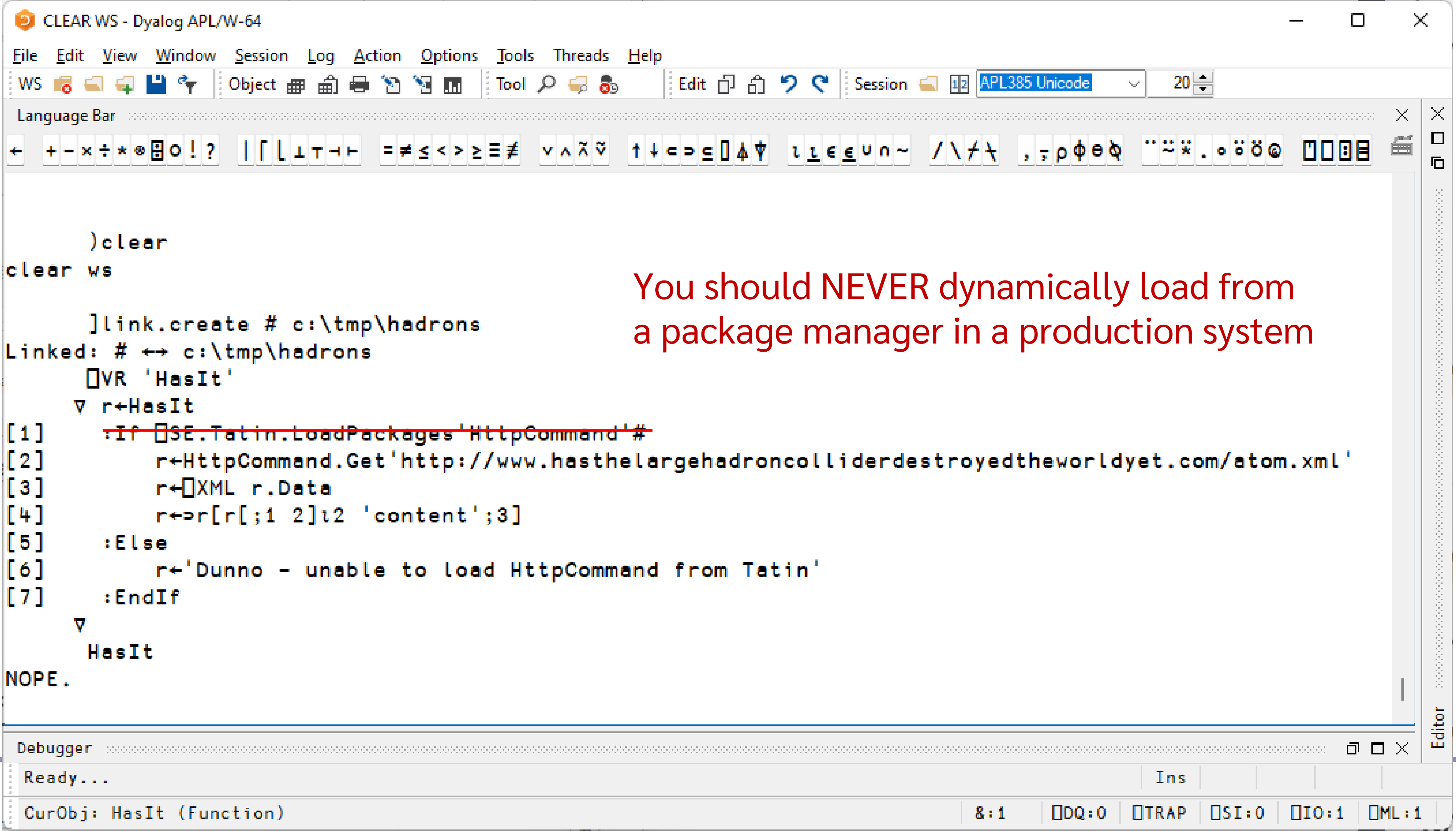
```
      )clear
clear ws

      ]link.create # c:\tmp\hadrons
Linked: # ↔ c:\tmp\hadrons
      ]Tatin.LoadPackages HttpCommand
Scanning 1 registry for HttpCommand...
   found in https://tatin.dev/
  Attempting to install https://tatin.dev/dyalog-HttpCommand-5.1.5...
  Establish dependencies...
  1 dependency identified
  Fetching from Tatin package cache...
  Add dyalog-HttpCommand-5.1.5 to dependency file...
  Saving build list...
  Establish references to dependencies...
1 package (including dependencies) loaded
      HasIt
NOPE.
```

Ready...   Ins

CurObj: HasIt (Function)   &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1



```
      )clear
clear ws

      ]link.create # c:\tmp\hadrons
Linked: # ↔ c:\tmp\hadrons
      ]Tatin.LoadPackages HttpCommand
Scanning 1 registry for HttpCommand...
   found in https://tatin.dev/
  Attempting to install https://tatin.dev/dyalog-HttpCommand-5.1.5...
  Establish dependencies...
  1 dependency identified
  Fetching from Tatin package cache...
  Add dyalog-HttpCommand-5.1.5 to dependency file...
  Saving build list...
  Establish references to dependencies...
1 package (including dependencies) loaded
      HasIt
NOPE.
```


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session
APL385 Unicode
20
Language Bar
Debugger
Editor

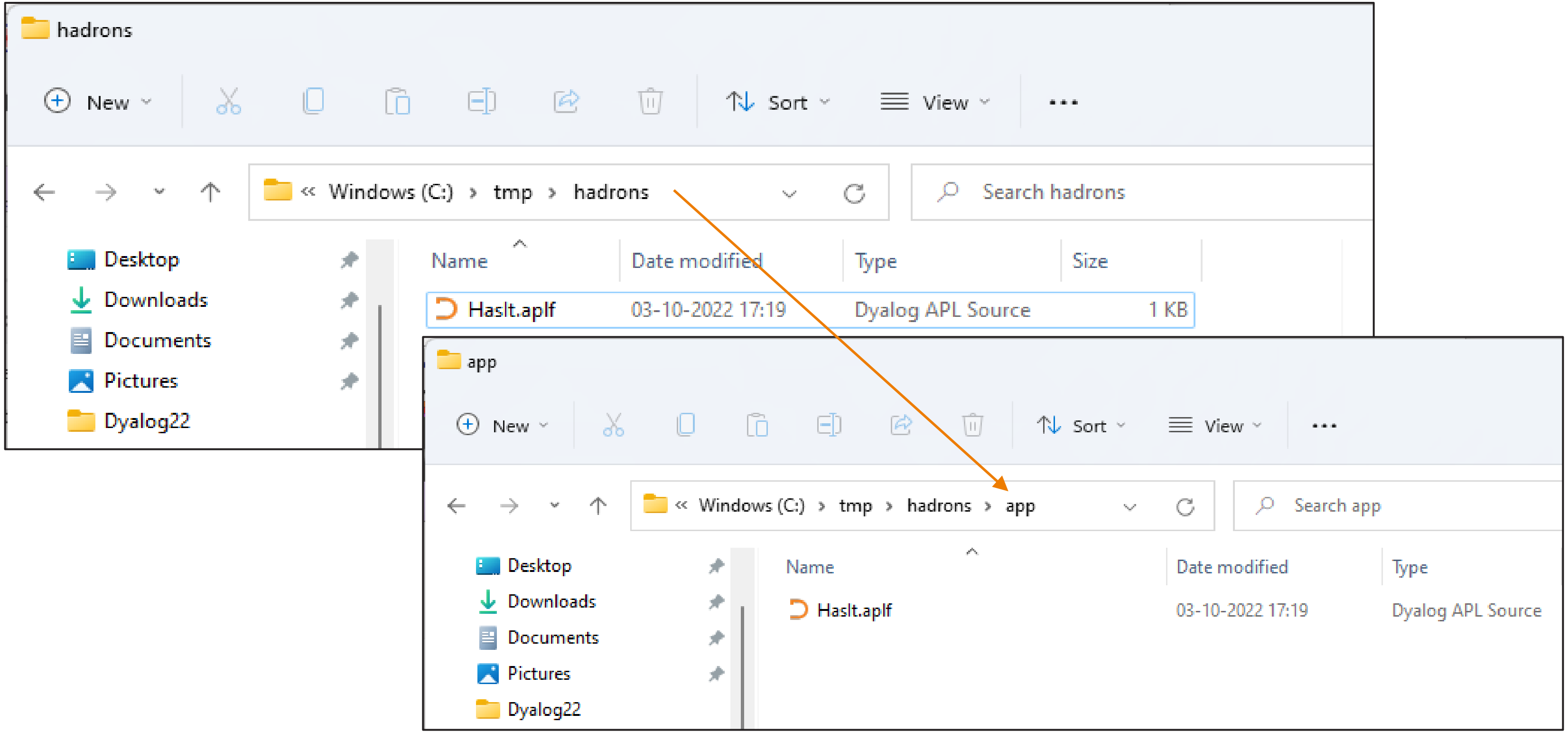
You should NEVER dynamically load from a package manager in a production system

```
      )clear
clear ws

      ]link.create # c:\tmp\hadrons
Linked: # ←→ c:\tmp\hadrons
      ⎕VR 'HasIt'
     ∇ r←HasIt
[1]    ~~:If ⎕SE.Tatin.LoadPackages'HttpCommand'#~~
[2]        r←HttpCommand.Get'http://www.hasthelargehadroncolliderdestroyedtheworldyet.com/atom.xml'
[3]        r←⎕XML r.Data
[4]        r←⊃r[r[;1 2]⍳2 'content';3]
[5]    :Else
[6]        r←'Dunno - unable to load HttpCommand from Tatin'
[7]    :EndIf
     ∇
      HasIt
NOPE.
```

Ready...   Ins

CurObj: HasIt (Function)   &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1

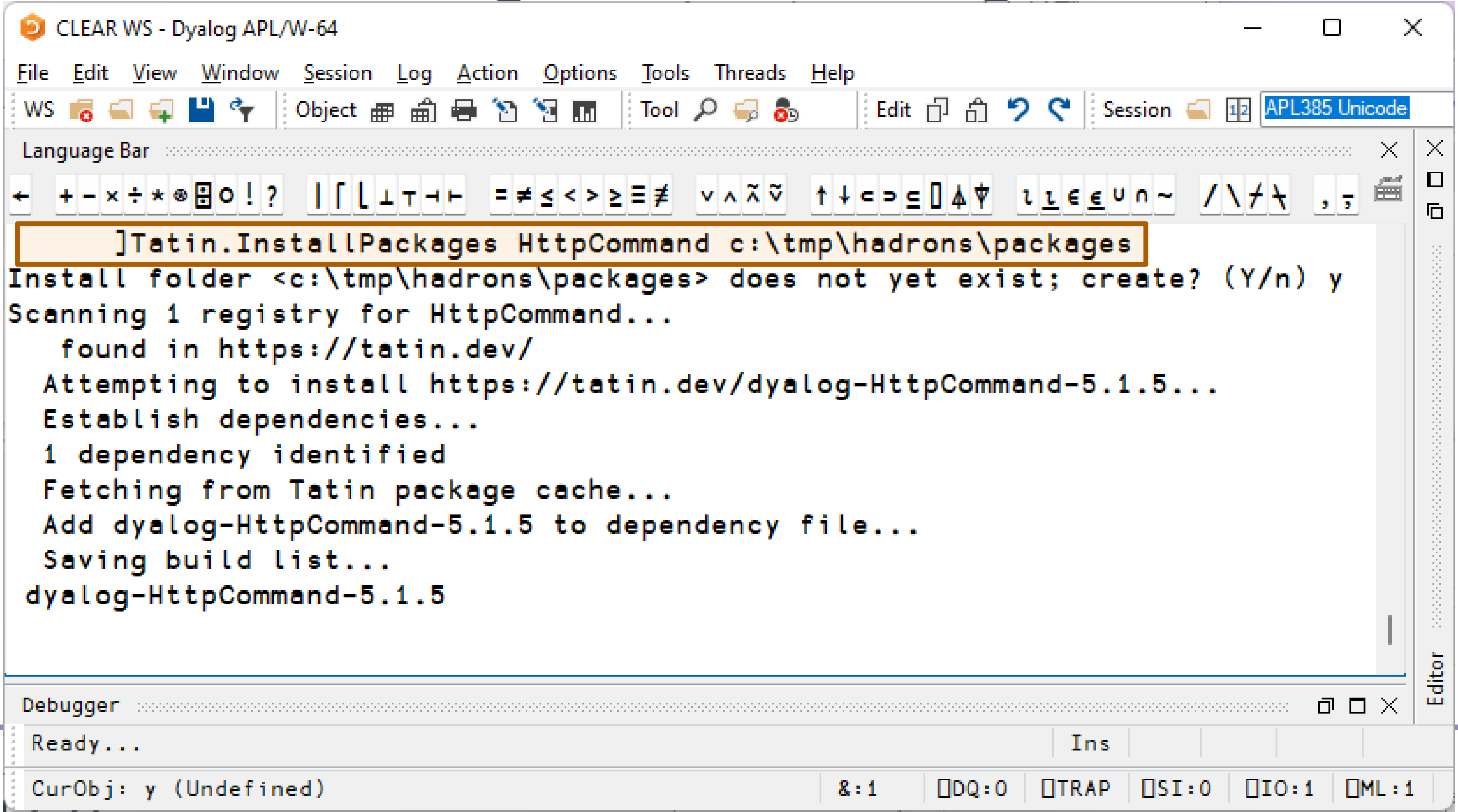
More Structure Needed



Name	Date modified	Type	Size
HasIt.aplf	03-10-2022 17:19	Dyalog APL Source	1 KB

Name	Date modified	Type
HasIt.aplf	03-10-2022 17:19	Dyalog APL Source

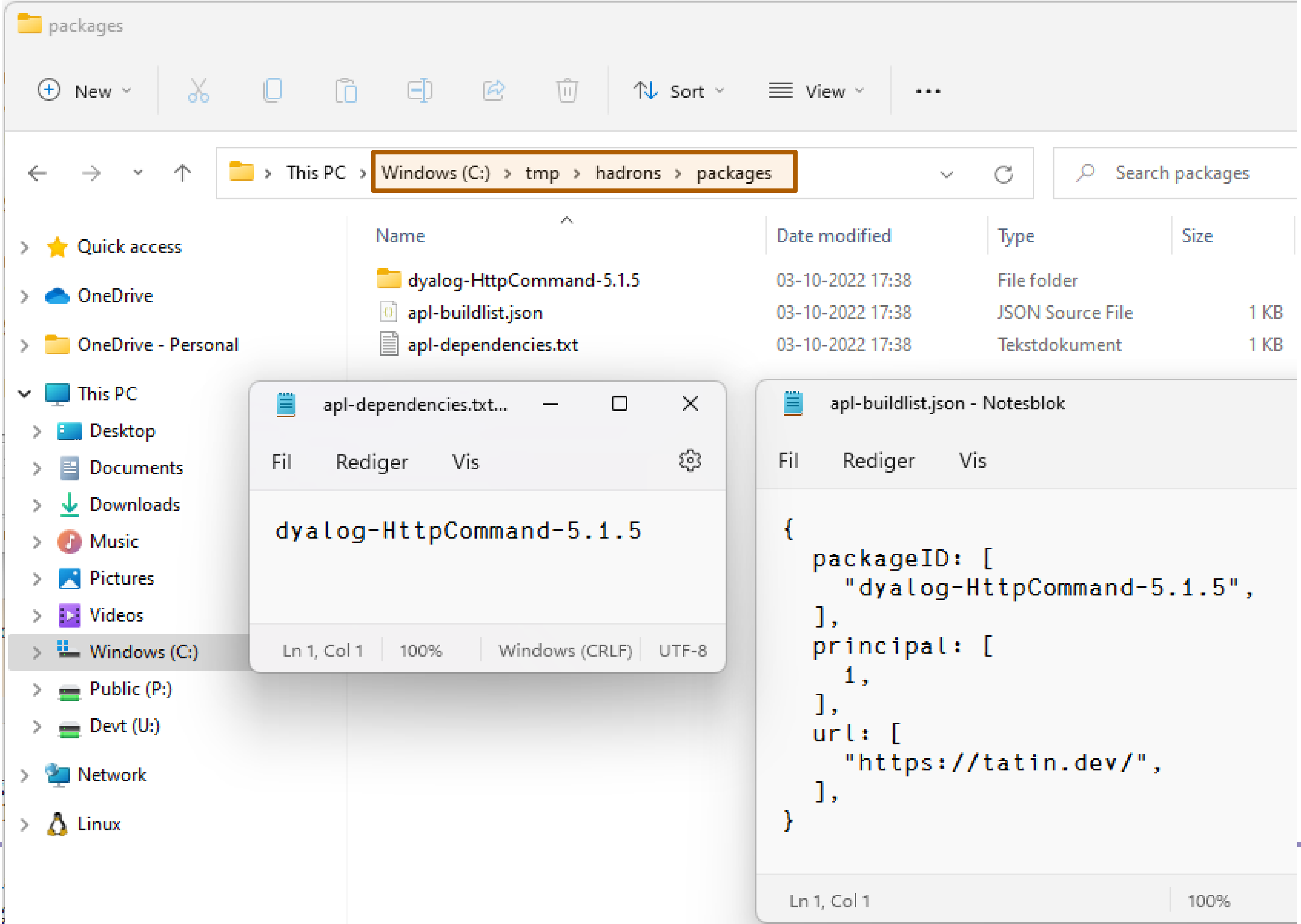
Install Package into sibling folder



CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL385 Unicode
Language Bar
Debugger
Editor

```
      ]Tatin.InstallPackages HttpCommand c:\tmp\hadrons\packages
Install folder <c:\tmp\hadrons\packages> does not yet exist; create? (Y/n) y
Scanning 1 registry for HttpCommand...
   found in https://tatin.dev/
  Attempting to install https://tatin.dev/dyalog-HttpCommand-5.1.5...
  Establish dependencies...
  1 dependency identified
  Fetching from Tatin package cache...
  Add dyalog-HttpCommand-5.1.5 to dependency file...
  Saving build list...
 dyalog-HttpCommand-5.1.5
```

```
Ready...                                    Ins
CurObj: y (Undefined)     &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```



This PC › Windows (C:) › tmp › hadrons › packages

Name	Date modified	Type	Size
dyalog-HttpCommand-5.1.5	03-10-2022 17:38	File folder	
apl-buildlist.json	03-10-2022 17:38	JSON Source File	1 KB
apl-dependencies.txt	03-10-2022 17:38	Tekstdokument	1 KB

apl-dependencies.txt...

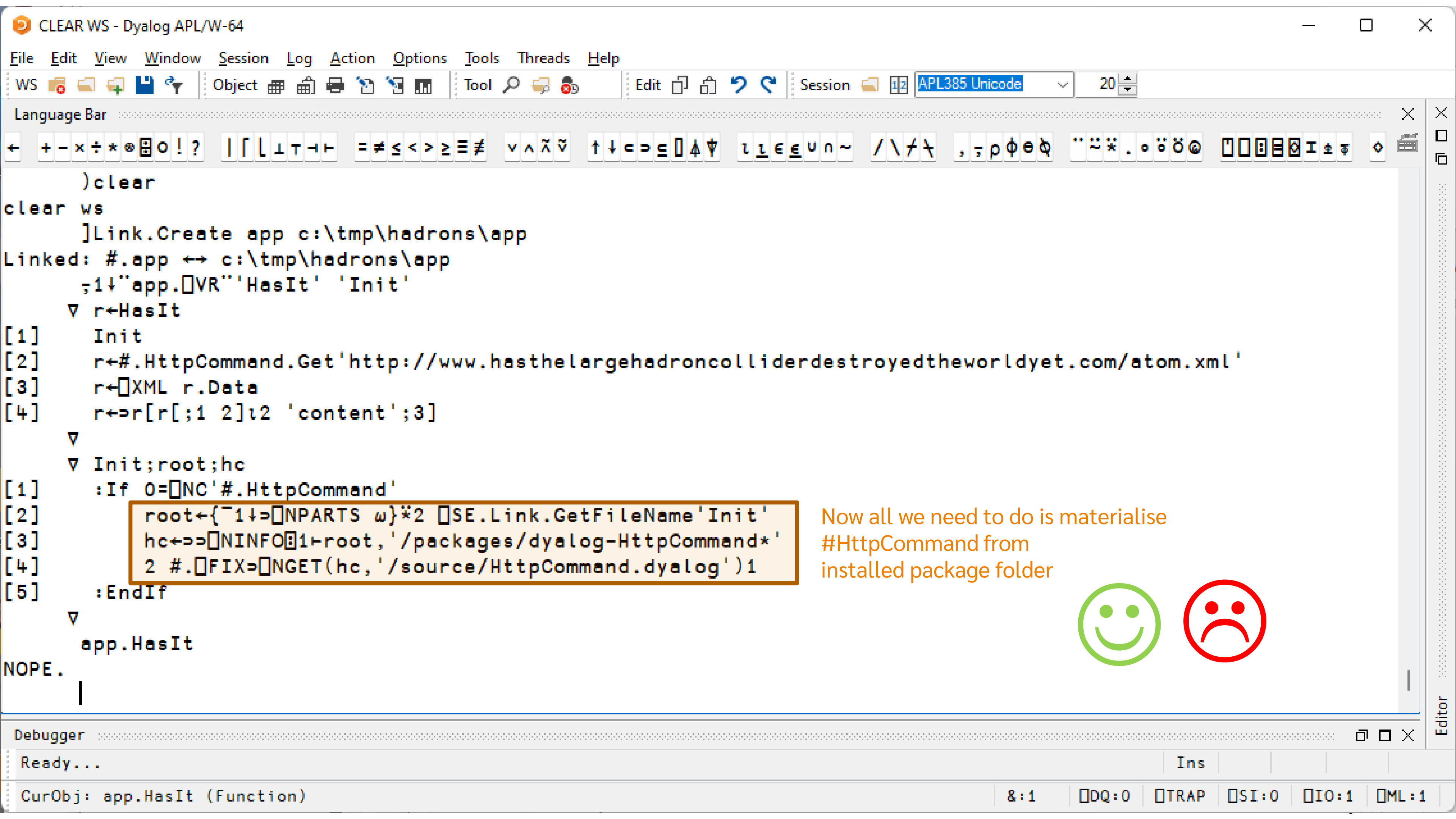
```
dyalog-HttpCommand-5.1.5
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

apl-buildlist.json - Notesblok

```
{
  packageID: [
    "dyalog-HttpCommand-5.1.5",
  ],
  principal: [
    1,
  ],
  url: [
    "https://tatin.dev/",
  ],
}
```

Ln 1, Col 1 | 100%


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
Editor

```
      )clear
clear ws
      ]Link.Create app c:\tmp\hadrons\app
Linked: #.app ↔ c:\tmp\hadrons\app
      ⍪1↓¨app.⎕VR¨'HasIt' 'Init'
     ∇ r←HasIt
[1]    Init
[2]    r←#.HttpCommand.Get'http://www.hasthelargehadroncolliderdestroyedtheworldyet.com/atom.xml'
[3]    r←⎕XML r.Data
[4]    r←⊃r[r[;1 2]⍳2 'content';3]
     ∇
     ∇ Init;root;hc
[1]    :If 0=⎕NC'#.HttpCommand'
[2]        root←{¯1↓⊃⎕NPARTS ⍵}⍣2 ⎕SE.Link.GetFileName'Init'
[3]        hc←⊃⊃⎕NINFO⍠1⊢root,'/packages/dyalog-HttpCommand*'
[4]        2 #.⎕FIX⊃⎕NGET(hc,'/source/HttpCommand.dyalog')1
[5]    :EndIf
     ∇
      app.HasIt
NOPE.
```

Now all we need to do is materialise #HttpCommand from installed package folder

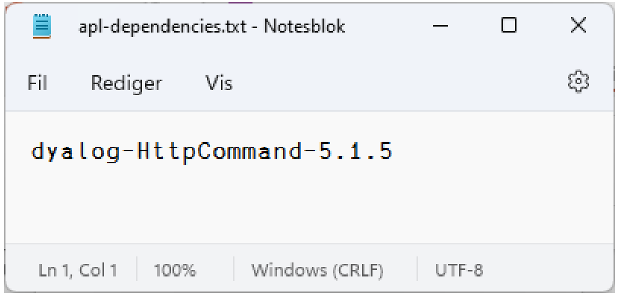
Ready...   Ins

CurObj: app.HasIt (Function)   &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1

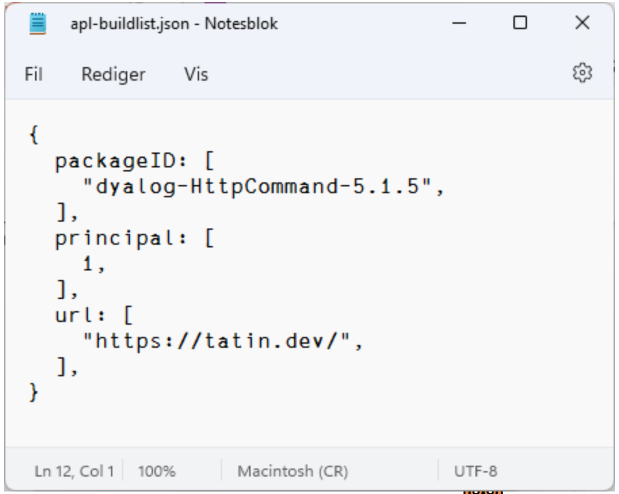
And really...

- ... we should inspect apl-dependencies.txt
- And loop through apl-buildlist

- This is what a "Project Manager" does
- Time for some Cider with that Tart



```
dyalog-HttpCommand-5.1.5
```



```
{
  packageID: [
    "dyalog-HttpCommand-5.1.5",
  ],
  principal: [
    1,
  ],
  url: [
    "https://tatin.dev/",
  ],
}
```

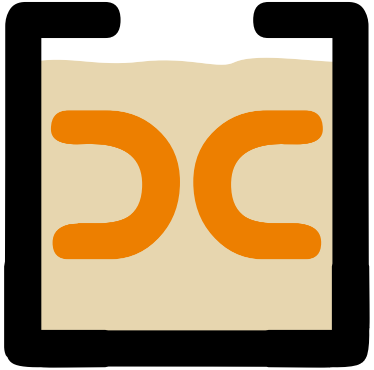
Another way to enjoy apples

Tatin

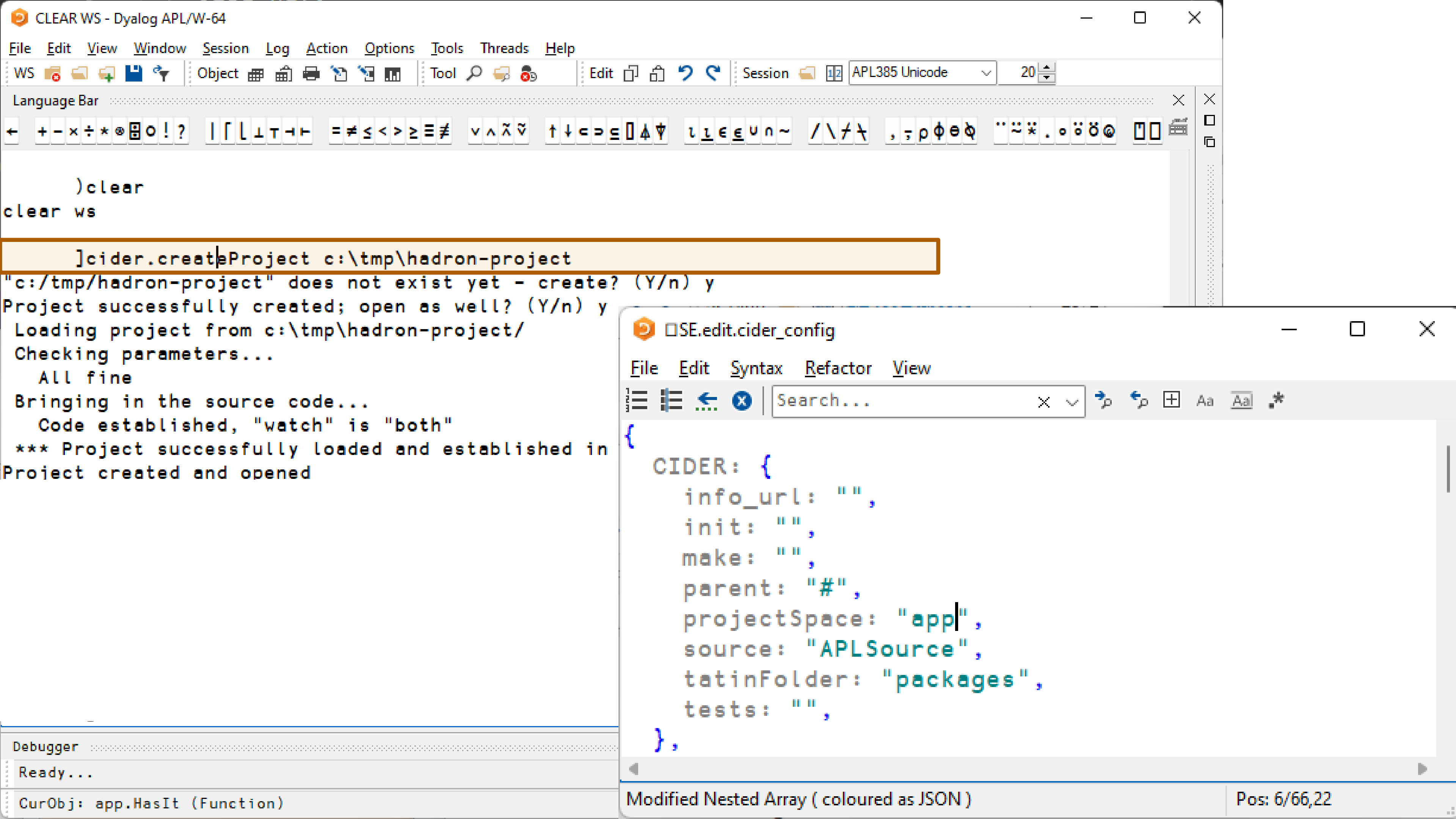
Cider

Another way to enjoy apples

Tatin



Cider



```
      )clear
clear ws

      ]cider.createProject c:\tmp\hadron-project
"c:/tmp/hadron-project" does not exist yet - create? (Y/n) y
Project successfully created; open as well? (Y/n) y
 Loading project from c:\tmp\hadron-project/
 Checking parameters...
   All fine
 Bringing in the source code...
   Code established, "watch" is "both"
 *** Project successfully loaded and established in
Project created and opened
```

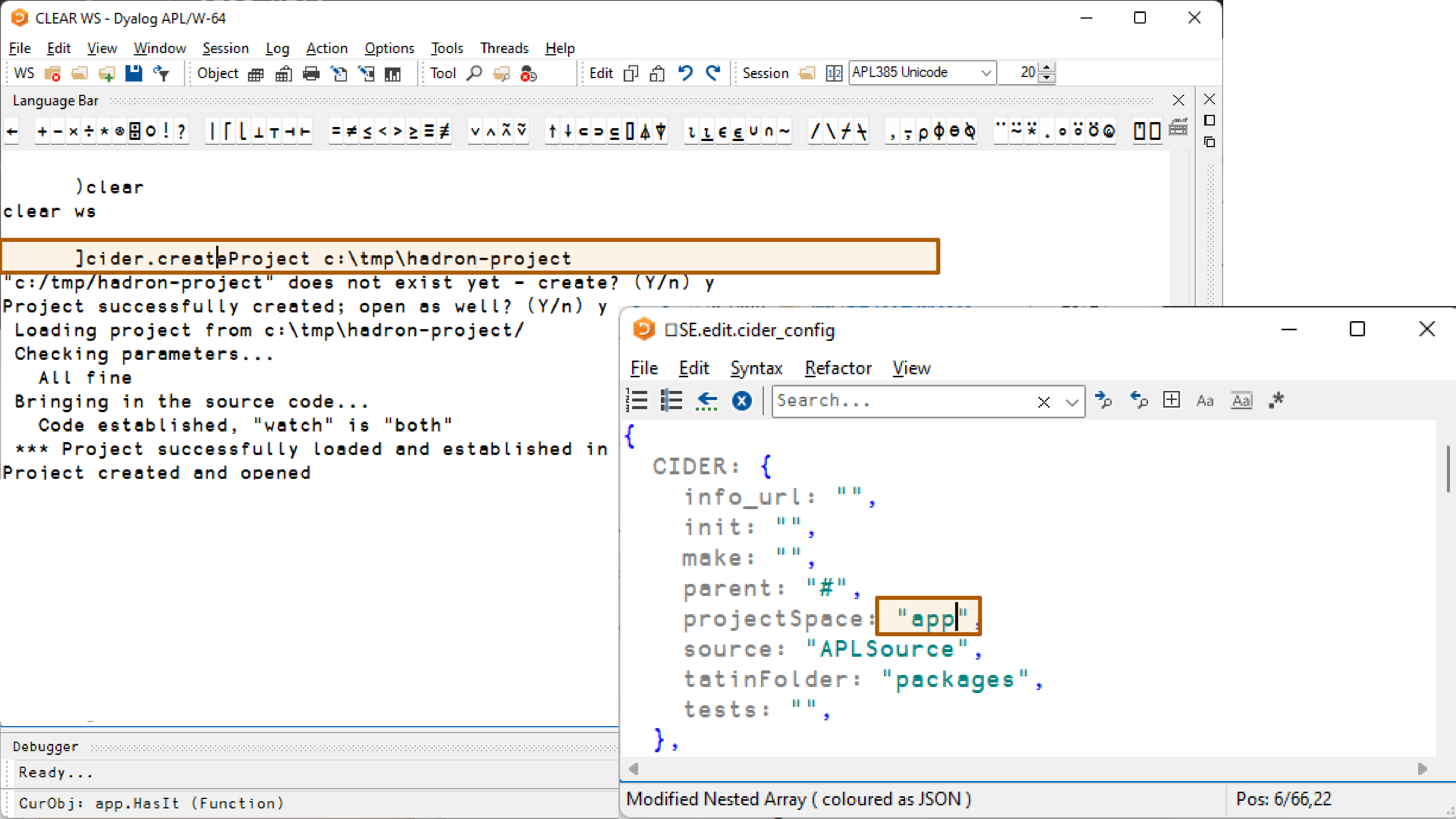
```
{
  CIDER: {
    info_url: "",
    init: "",
    make: "",
    parent: "#",
    projectSpace: "app",
    source: "APLSource",
    tatinFolder: "packages",
    tests: "",
  },
```

Ready...

CurObj: app.HasIt (Function)

Modified Nested Array (coloured as JSON)

Pos: 6/66,22

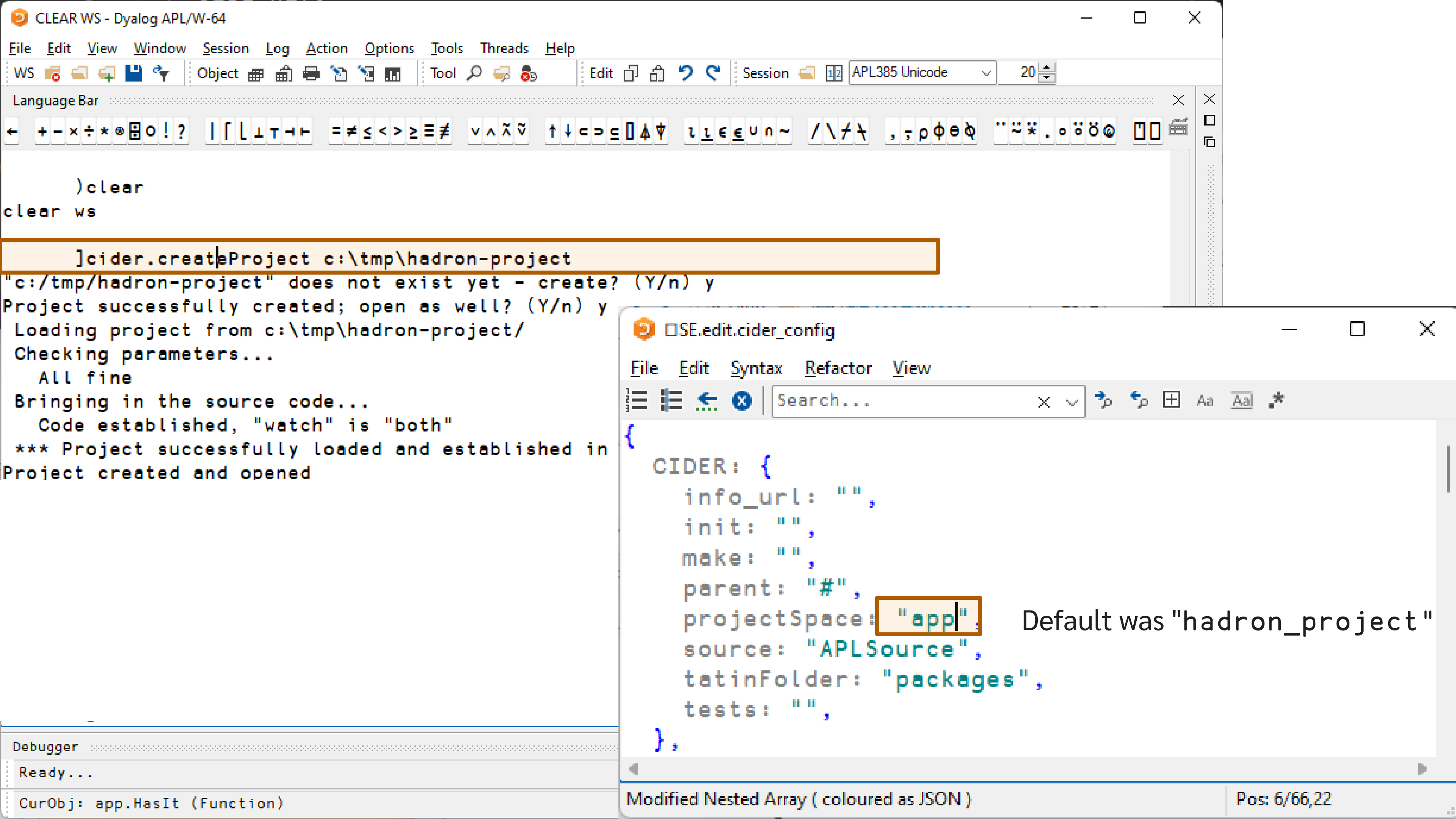

CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
⎕SE.edit.cider_config
File Edit Syntax Refactor View
Search...

```
      )clear
clear ws

      ]cider.createProject c:\tmp\hadron-project
"c:/tmp/hadron-project" does not exist yet - create? (Y/n) y
Project successfully created; open as well? (Y/n) y
 Loading project from c:\tmp\hadron-project/
 Checking parameters...
   All fine
 Bringing in the source code...
   Code established, "watch" is "both"
 *** Project successfully loaded and established in
Project created and opened
```

```
{
  CIDER: {
    info_url: "",
    init: "",
    make: "",
    parent: "#",
    projectSpace: "app",
    source: "APLSource",
    tatinFolder: "packages",
    tests: "",
  },
```

Modified Nested Array (coloured as JSON)   Pos: 6/66,22

```
Ready...
CurObj: app.HasIt (Function)
```


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
⎕SE.edit.cider_config
File Edit Syntax Refactor View
Search...

```
      )clear
clear ws

      ]cider.createProject c:\tmp\hadron-project
"c:/tmp/hadron-project" does not exist yet - create? (Y/n) y
Project successfully created; open as well? (Y/n) y
 Loading project from c:\tmp\hadron-project/
 Checking parameters...
   All fine
 Bringing in the source code...
   Code established, "watch" is "both"
 *** Project successfully loaded and established in
Project created and opened
```

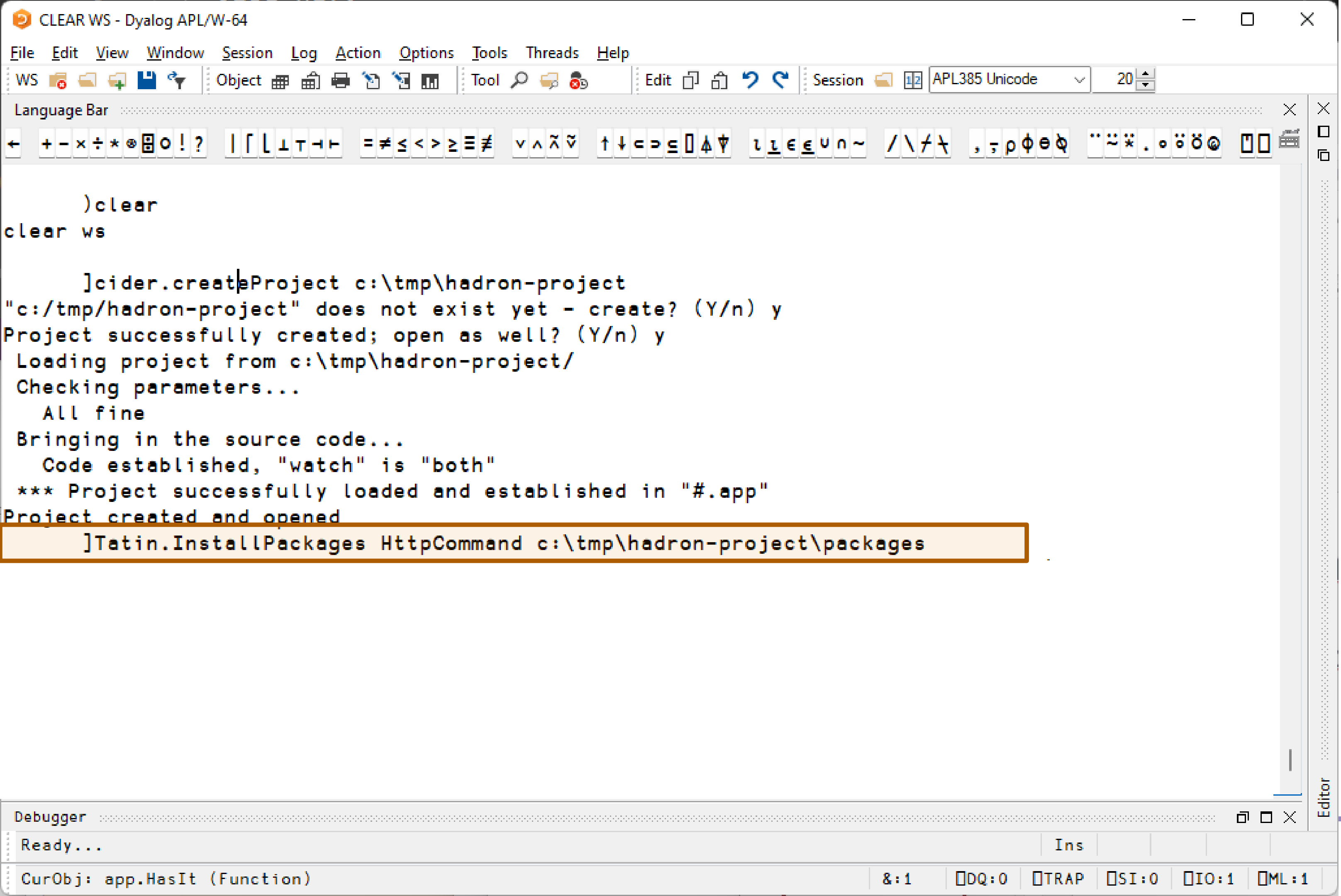
```
Debugger
 Ready...
 CurObj: app.HasIt (Function)
```

```
{
  CIDER: {
    info_url: "",
    init: "",
    make: "",
    parent: "#",
    projectSpace: "app",
    source: "APLSource",
    tatinFolder: "packages",
    tests: "",
  },
```

Default was "hadron_project"

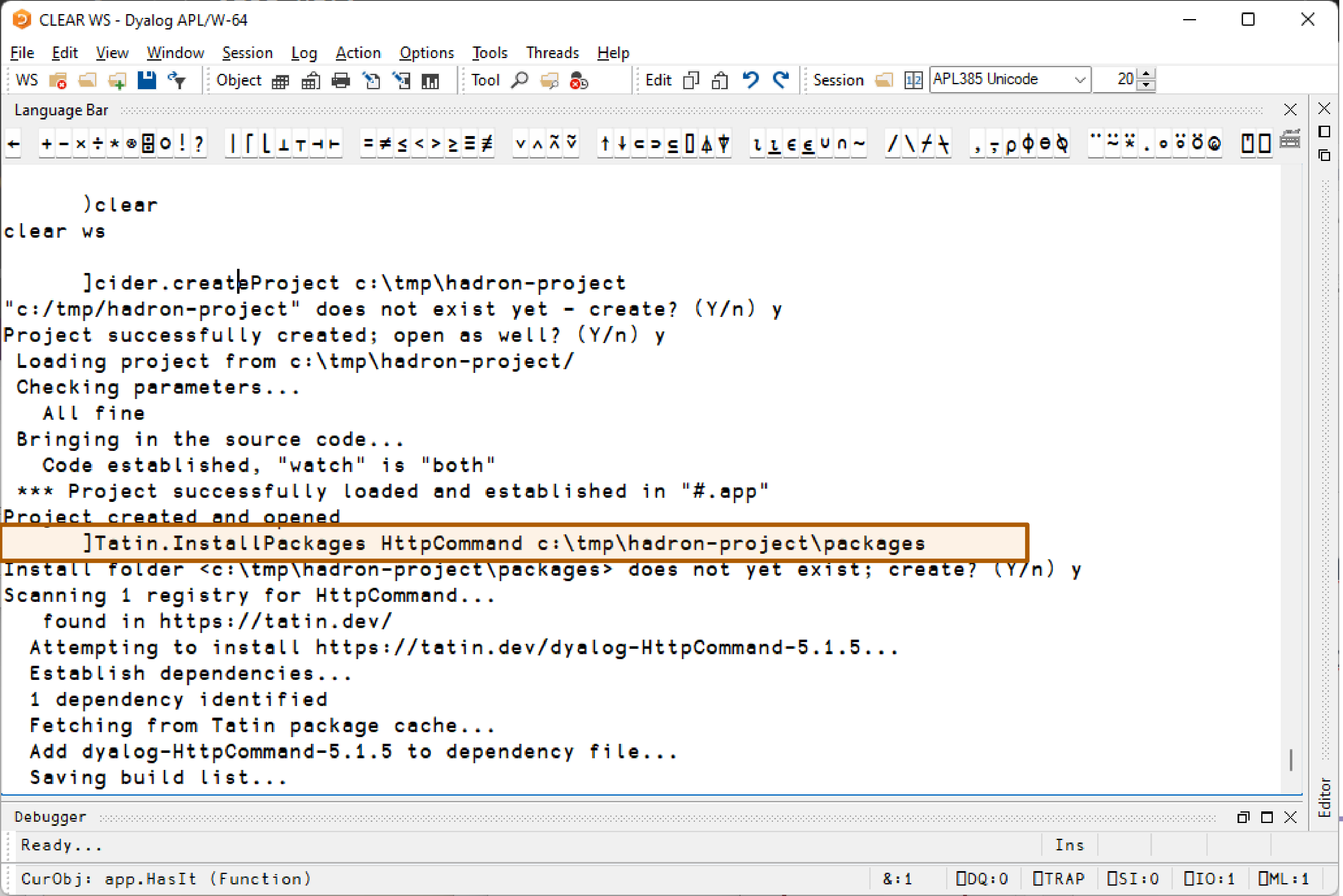
Modified Nested Array (coloured as JSON)

Pos: 6/66,22


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS
Object
Tool
Edit
Session
APL385 Unicode
20
Language Bar
Debugger
Editor

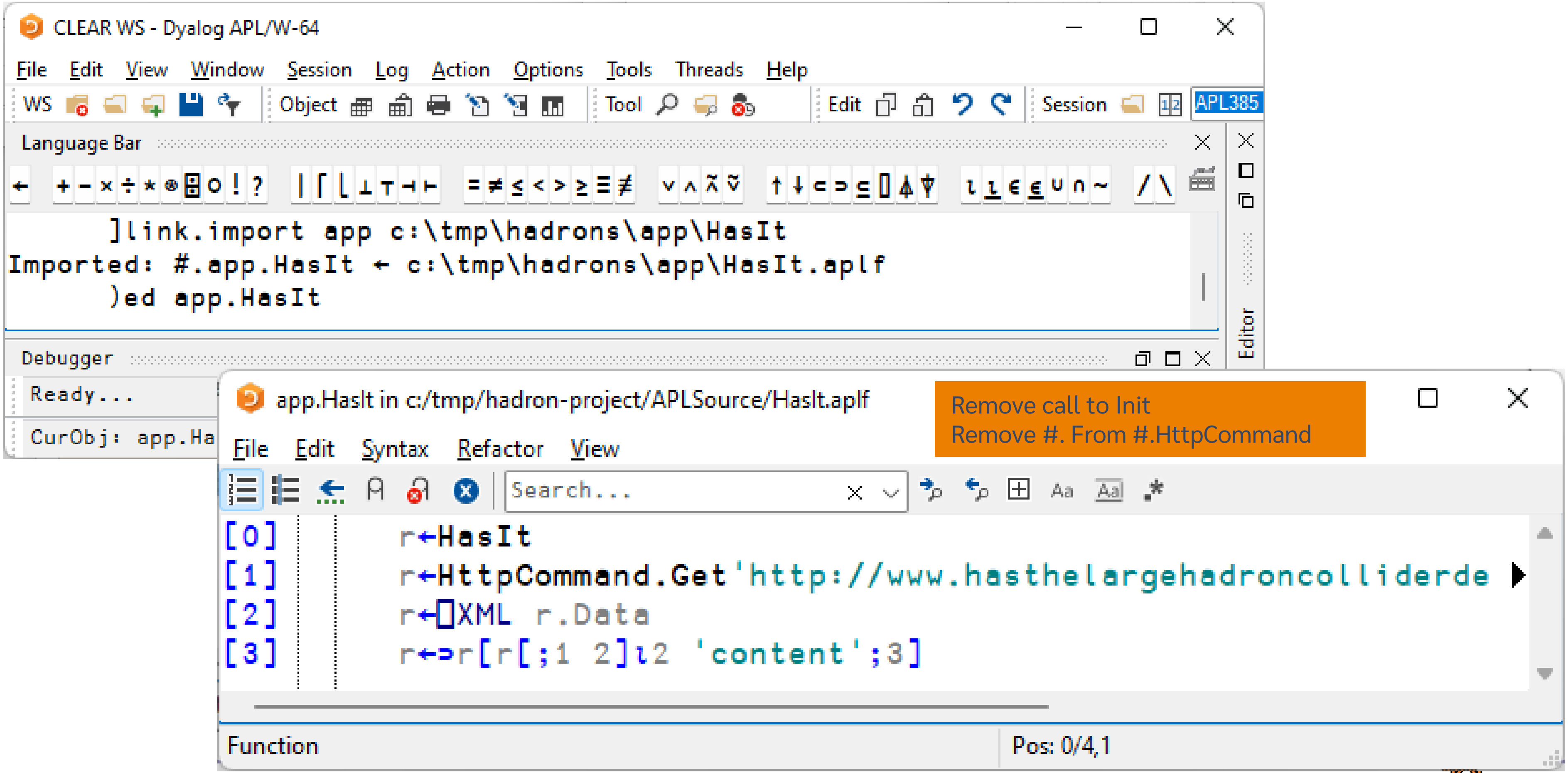
```
      )clear
clear ws

      ]cider.createProject c:\tmp\hadron-project
"c:/tmp/hadron-project" does not exist yet - create? (Y/n) y
Project successfully created; open as well? (Y/n) y
 Loading project from c:\tmp\hadron-project/
 Checking parameters...
   All fine
 Bringing in the source code...
   Code established, "watch" is "both"
 *** Project successfully loaded and established in "#.app"
Project created and opened
      ]Tatin.InstallPackages HttpCommand c:\tmp\hadron-project\packages
```

```
Ready...                                                    Ins
CurObj: app.HasIt (Function)      &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```



```
      )clear
clear ws

      ]cider.createProject c:\tmp\hadron-project
"c:/tmp/hadron-project" does not exist yet - create? (Y/n) y
Project successfully created; open as well? (Y/n) y
 Loading project from c:\tmp\hadron-project/
 Checking parameters...
   All fine
 Bringing in the source code...
   Code established, "watch" is "both"
 *** Project successfully loaded and established in "#.app"
Project created and opened
      ]Tatin.InstallPackages HttpCommand c:\tmp\hadron-project\packages
Install folder <c:\tmp\hadron-project\packages> does not yet exist; create? (Y/n) y
Scanning 1 registry for HttpCommand...
   found in https://tatin.dev/
  Attempting to install https://tatin.dev/dyalog-HttpCommand-5.1.5...
  Establish dependencies...
  1 dependency identified
  Fetching from Tatin package cache...
  Add dyalog-HttpCommand-5.1.5 to dependency file...
  Saving build list...
```

```
Ready...                                   Ins
CurObj: app.HasIt (Function)    &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```

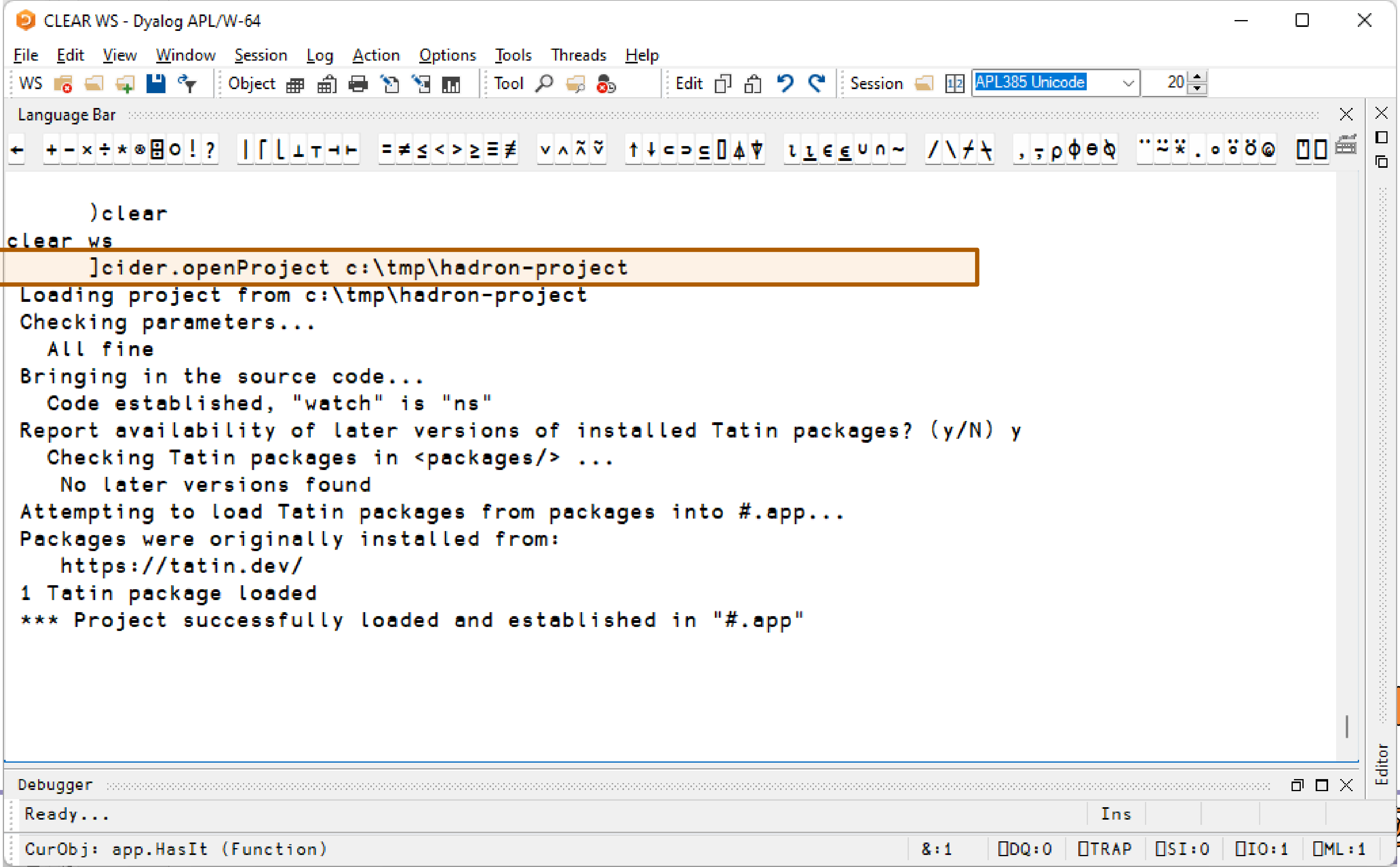

CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options Tools Threads Help
WS Object Tool Edit Session APL385
Language Bar
Debugger
Ready...
CurObj: app.Ha
Editor
File Edit Syntax Refactor View
Search...
Function
Pos: 0/4,1

```
      ]link.import app c:\tmp\hadrons\app\HasIt
Imported: #.app.HasIt ← c:\tmp\hadrons\app\HasIt.aplf
      )ed app.HasIt
```

app.HasIt in c:/tmp/hadron-project/APLSource/HasIt.aplf

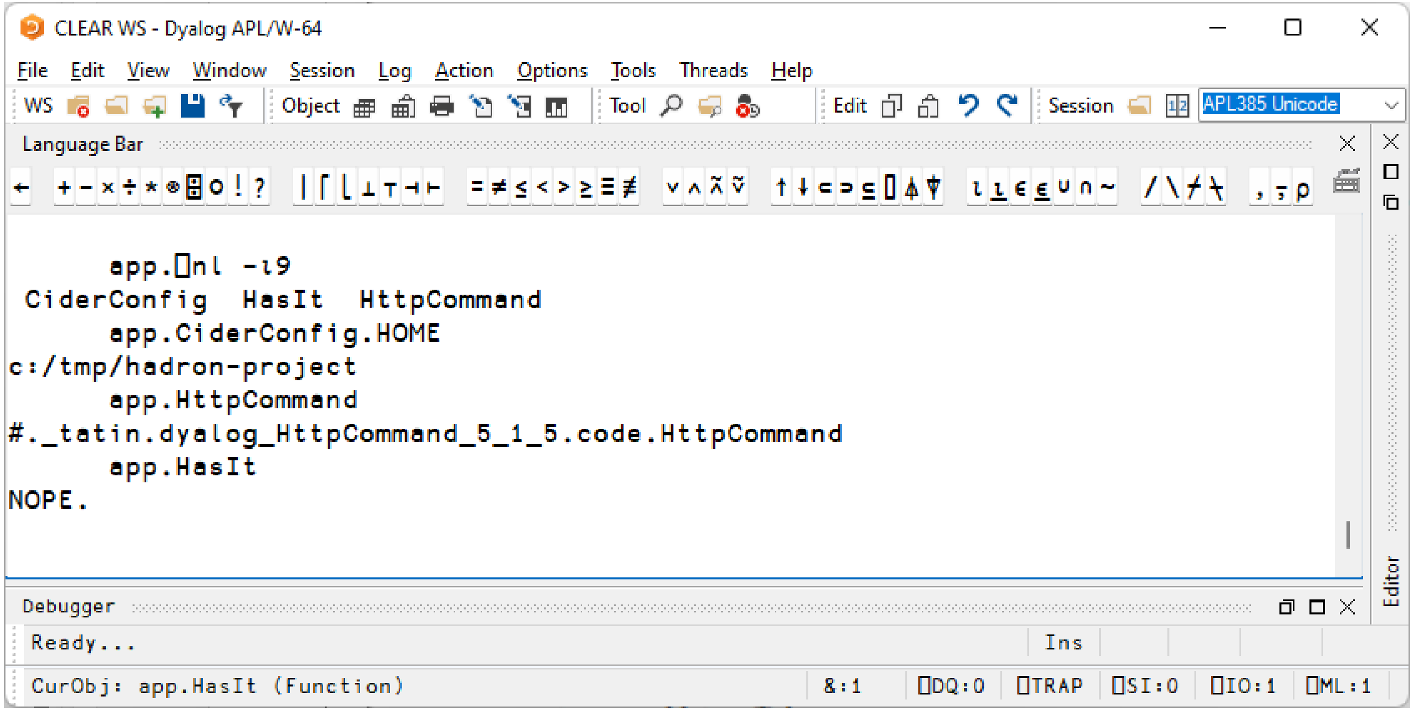
```
      r←HasIt
      r←HttpCommand.Get'http://www.hasthelargehadroncolliderde
      r←⎕XML r.Data
      r←⊃r[r[;1 2]⍳2 'content';3]
```

Remove call to Init
Remove #. From #.HttpCommand



```
      )clear
clear ws
      ]cider.openProject c:\tmp\hadron-project
 Loading project from c:\tmp\hadron-project
 Checking parameters...
   All fine
 Bringing in the source code...
   Code established, "watch" is "ns"
 Report availability of later versions of installed Tatin packages? (y/N) y
   Checking Tatin packages in <packages/> ...
    No later versions found
 Attempting to load Tatin packages from packages into #.app...
 Packages were originally installed from:
    https://tatin.dev/
 1 Tatin package loaded
 *** Project successfully loaded and established in "#.app"
```

```
Ready...                                                    Ins
CurObj: app.HasIt (Function)           &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```

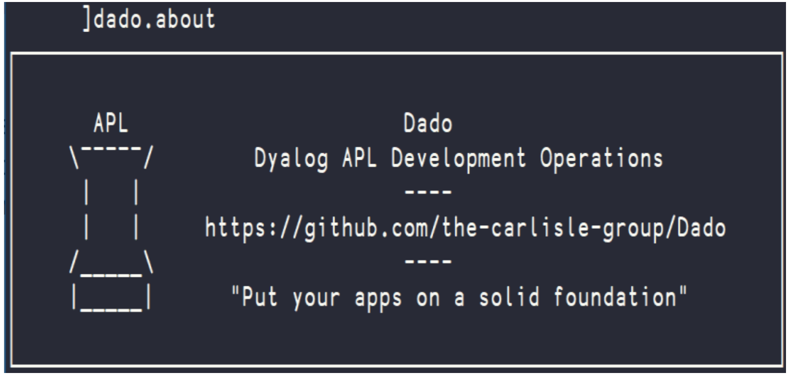


```
      app.⎕nl -⍳9
 CiderConfig  HasIt  HttpCommand
      app.CiderConfig.HOME
c:/tmp/hadron-project
      app.HttpCommand
#._tatin.dyalog_HttpCommand_5_1_5.code.HttpCommand
      app.HasIt
NOPE.
```

```
Ready...                                      Ins
CurObj: app.HasIt (Function)      &:1   ⎕DQ:0   ⎕TRAP   ⎕SI:0   ⎕IO:1   ⎕ML:1
```

Acre and Dado

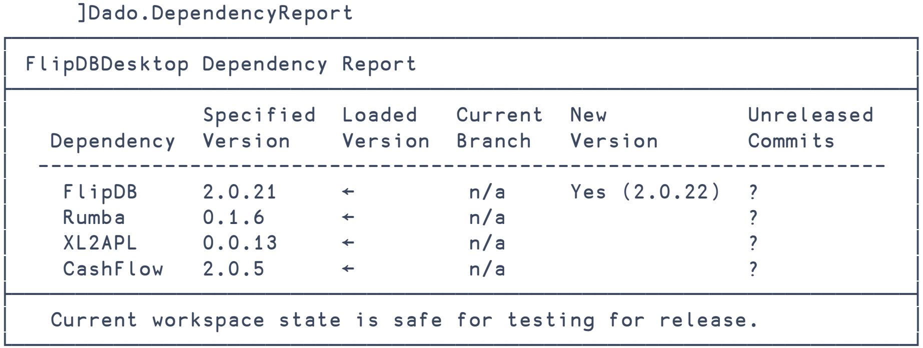
- Acre: In use for >10 years
  - Originally used component files
  - Recent versions based on text source
  - Stable foundation for Dado
- Dado: Adds Git workflow to Acre
  - Manages dependencies between Dado projects
  - Previously presented by Josh David in Dyalog'21 talk
  - Foundations of workflow presented by Paul Mansour at Dyalog'19



```
      ]dado.about

   APL                       Dado
 \-----/        Dyalog APL Development Operations
  |   |                      ----
  |   |      https://github.com/the-carlisle-group/Dado
 /_____\                     ----
 |_____|        "Put your apps on a solid foundation"
```

Benefits of Dado

- Direct support for Git[Hub]
  - No need to chose a Git workflow, Dado already chose one
- Dependency management
- Automatic Release generation
  - Dado will automatically create and format release notes based on commit history
- Accessible from the APL Session
  - No need to download other software to do git operations.



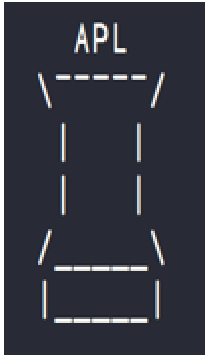
```
      ]Dado.DependencyReport
┌──────────────────────────────────────────────────────────────────────────────┐
│FlipDBDesktop Dependency Report                                               │
├──────────────────────────────────────────────────────────────────────────────┤
│            Specified  Loaded   Current  New            Unreleased            │
│  Dependency Version    Version  Branch   Version        Commits              │
│  ------------------------------------------------------------------------    │
│   FlipDB    2.0.21     ←          n/a    Yes (2.0.22)   ?                    │
│   Rumba     0.1.6      ←          n/a                   ?                    │
│   XL2APL    0.0.13     ←          n/a                   ?                    │
│   CashFlow  2.0.5      ←          n/a                   ?                    │
├──────────────────────────────────────────────────────────────────────────────┤
│  Current workspace state is safe for testing for release.                    │
└──────────────────────────────────────────────────────────────────────────────┘
```

Downsides of Dado

Dado is "opinionated": Perfect if the choices are right for you, but:

- Dado requires Git[Hub/Lab]
- Only supports one Git workflow
  - Not ideal for teams maintaining multiple versions of a project
- Only supports other Dado projects as dependencies

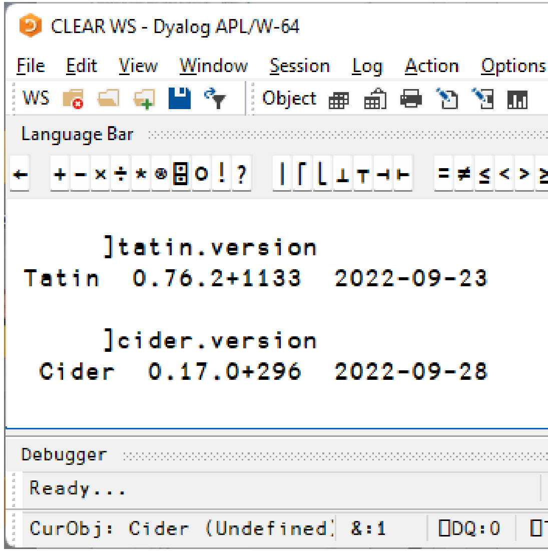
Dyalog recommends Dado, but is designing Cider as a more agnostic, extendable tool to complement Dado



Current Status of Tatin and Cider

- Tatin is maturing
  - Has seen significant testing (thanks, Davin!)
  - A useful number of packages exist
  - We hope to ship Tatin v1.0 with v19.0

- Cider is in it's infancy
  - Developed by Kai for own use during Tatin project
  - Already useful, but needs a few more cycles
  - We plan to make rapid progress on Cider this winter
    - Dyalog will participate in development


CLEAR WS - Dyalog APL/W-64
File Edit View Window Session Log Action Options
WS
Object
Language Bar
← + - × ÷ * ⍟ ⌹ ○ ! ? | ⌈ ⌊ ⊥ ⊤ ⊣ ⊢ = ≠ ≤ < >
Debugger
Ready...
CurObj: Cider (Undefined) &:1 ⎕DQ:0

```
      ]tatin.version
 Tatin  0.76.2+1133  2022-09-23

      ]cider.version
  Cider  0.17.0+296  2022-09-28
```

Things I'd like to see in Cider v1.0

- Support for many dependency types, for example

```
]cider.add-dependency tatin  HttpCommand
]cider.add-dependency dado   https://github.com/user/dado-pkg
]cider.add-dependency folder \devt\tools\some-internal-tool
]cider.add-dependency nuget  System.Drawing.Common
```

- Support for `]dbuild` and other build and/or deployment tools
    - Including very easy deployment to Tatin
- Support for `]dtest` and other test frameworks
- Support for "Assets", supporting multiple target platforms
- Management of configuration files for the interpreter and application
- An extensible architecture, so users / the community / can add new types of dependencies, build tools, new test frameworks, etc

Further ideas to support Projects

- "Open With Dyalog" should be able to identify whether a folder is
  - A Cider project
  - A Dado or Acre project
  - A collection of source files to be opened with Link (in the future, directly from the interpreter without using Link)
- An alternative definition of "Workspace" could be a configuration file which defines...
  - A collection of projects to be opened
  - Tools to be loaded for use during development
  - IDE configuration settings for the workspace (session captions, etc)

Dyalog:
Making APL more enjoyable

Tatin

Cider

Dyalog:
Making APL more enjoyable

Tatin

Cider

Dyalog:
Making APL more enjoyable