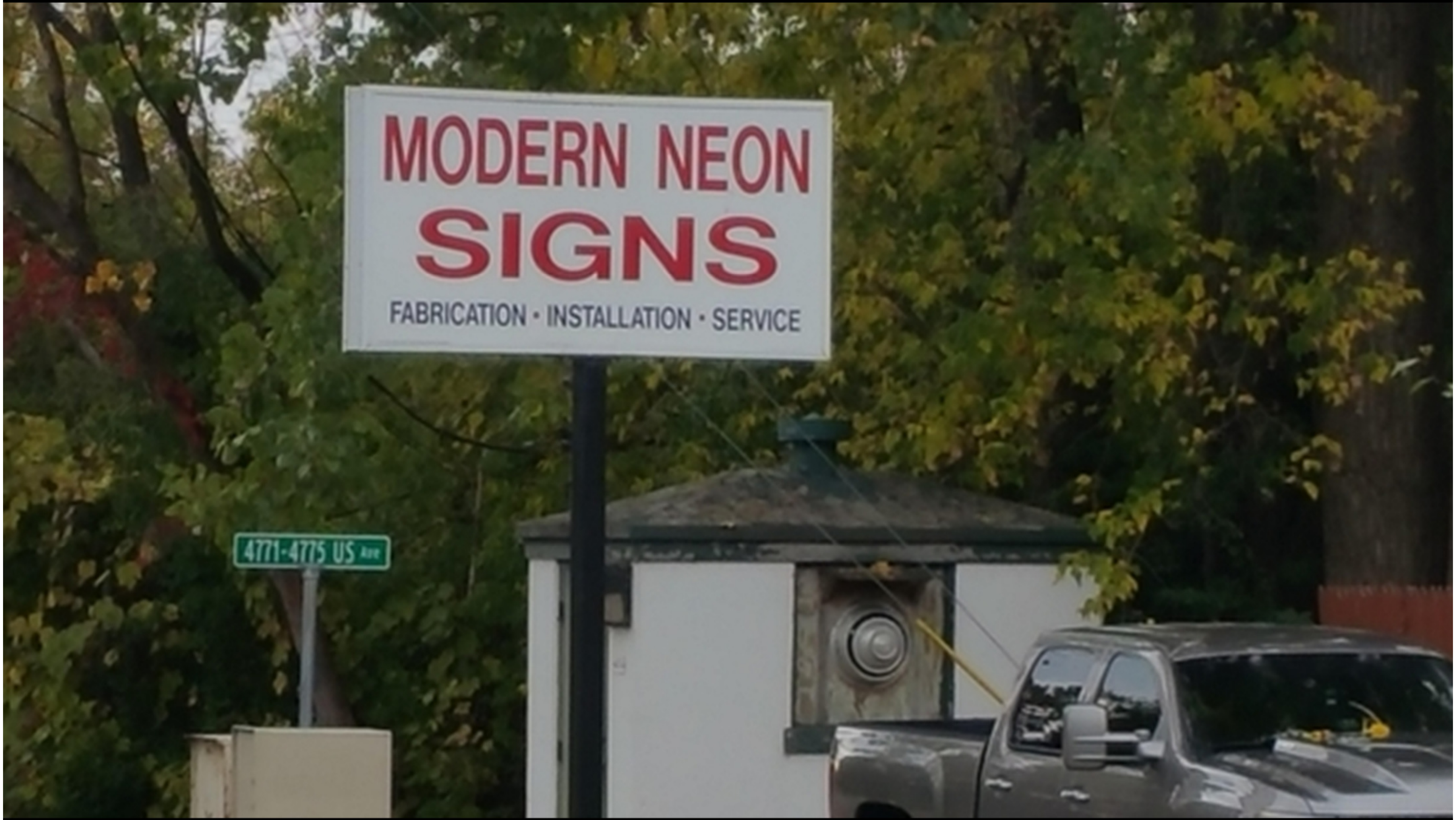
Missed Opportunity?



MOVE FORWARD ▸ TOGETHER

Scripta

Script Management Product

Capitalizing on Missed Opportunities

TPFUG 2019, Denver

Definitions

- What do we mean by “script” in this presentation?
  - A sequence of messages or entries introduced into the system to facilitate testing
  - Examples:
    - Regression scripts
    - Input scripts

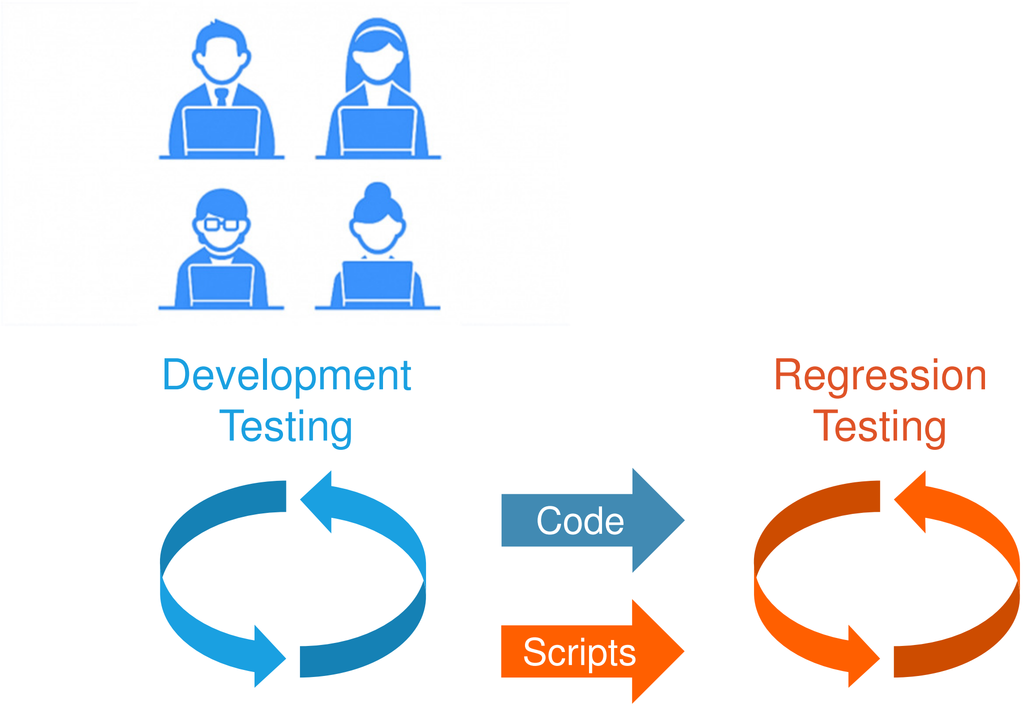
Why Scripta?

Missed opportunities for…

- **More synergy** between Development and Regression
- More **script reuse** by developers
- More **automation** in gathering candidate scripts
- More **varied types** of messages/entries in scripts
- **Greater efficiency in editing** scripts

Synergy Between Development and Regression

- Ideally, developers' scripts would regularly get submitted for consideration for adaptation in Regression



Development Testing

Code

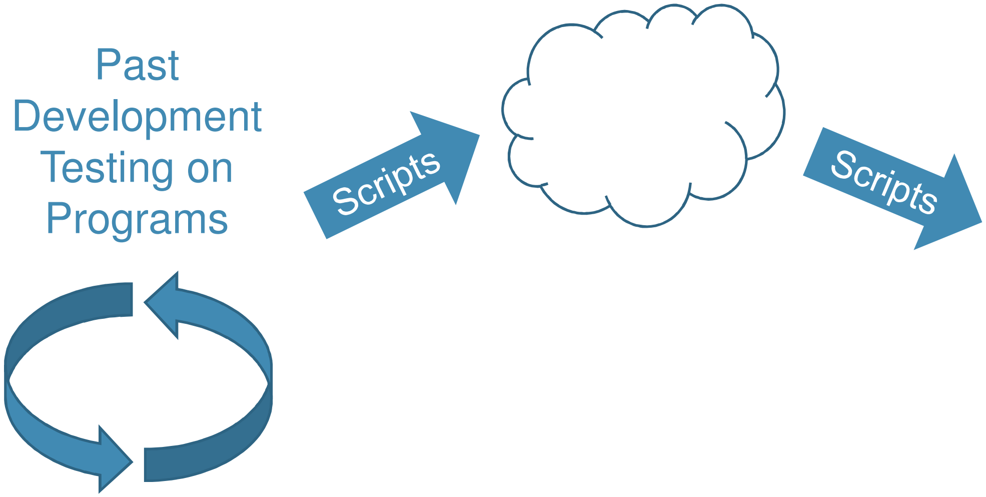
Scripts

Regression Testing

Problem: Developers have resisted script submission for varied reasons

More Script Reuse by Developers

- Ideally developers could easily find scripts – from present and past



Past Development Testing on Programs

Scripts

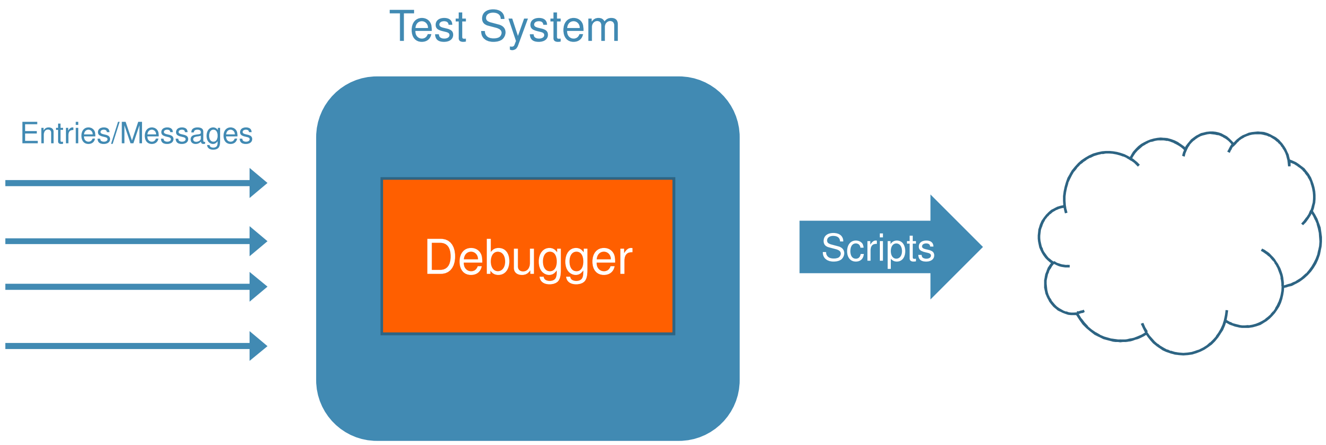
Scripts

Present Development Testing on Programs

Problem: Script reuse in Development may be dependent on individual organizational skills and memory

More Automation in Gathering Candidate Scripts

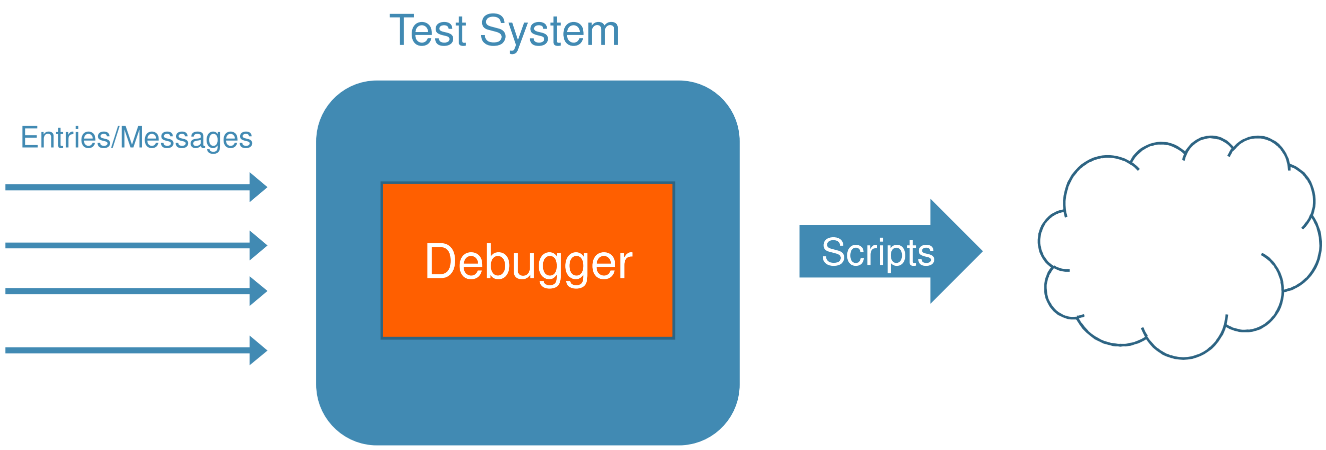
- All messages/entries into test systems are potential data for scripts



Problem: The actual messages and entries at this level are not easily reused

More Types of Messages/Entries in Scripts

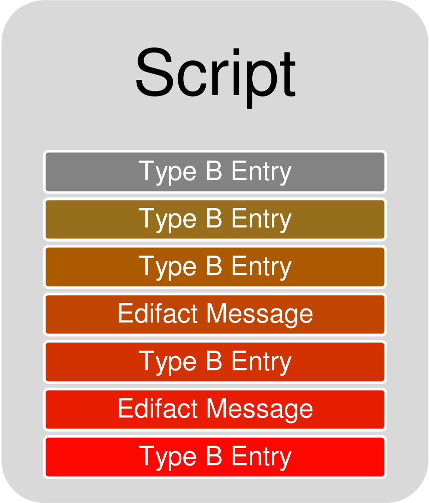
- Pay attention to all types of messages coming into the system
- Allow all types of messages to be included in scripts



Problem: The types of messages used for testing don't produce as comprehensive a suite of tests as desired

Greater Efficiency in Editing and Preparing Scripts

- Support for special features -- date replacement, variables, etc.
- Ideally, excellent support for script editing tasks would be available



Problem: Scripts and their advanced features are not easy to edit

What Is Scripta?

- ▶ Scripta Definition
- ▶ Architecture
- ▶ Database and Repository
- ▶ Scripts
- ▶ Editor
- ▶ Search
- ▶ Developer Workflow
- ▶ Regression Workflow

Scripta Definition

A Software Product for…

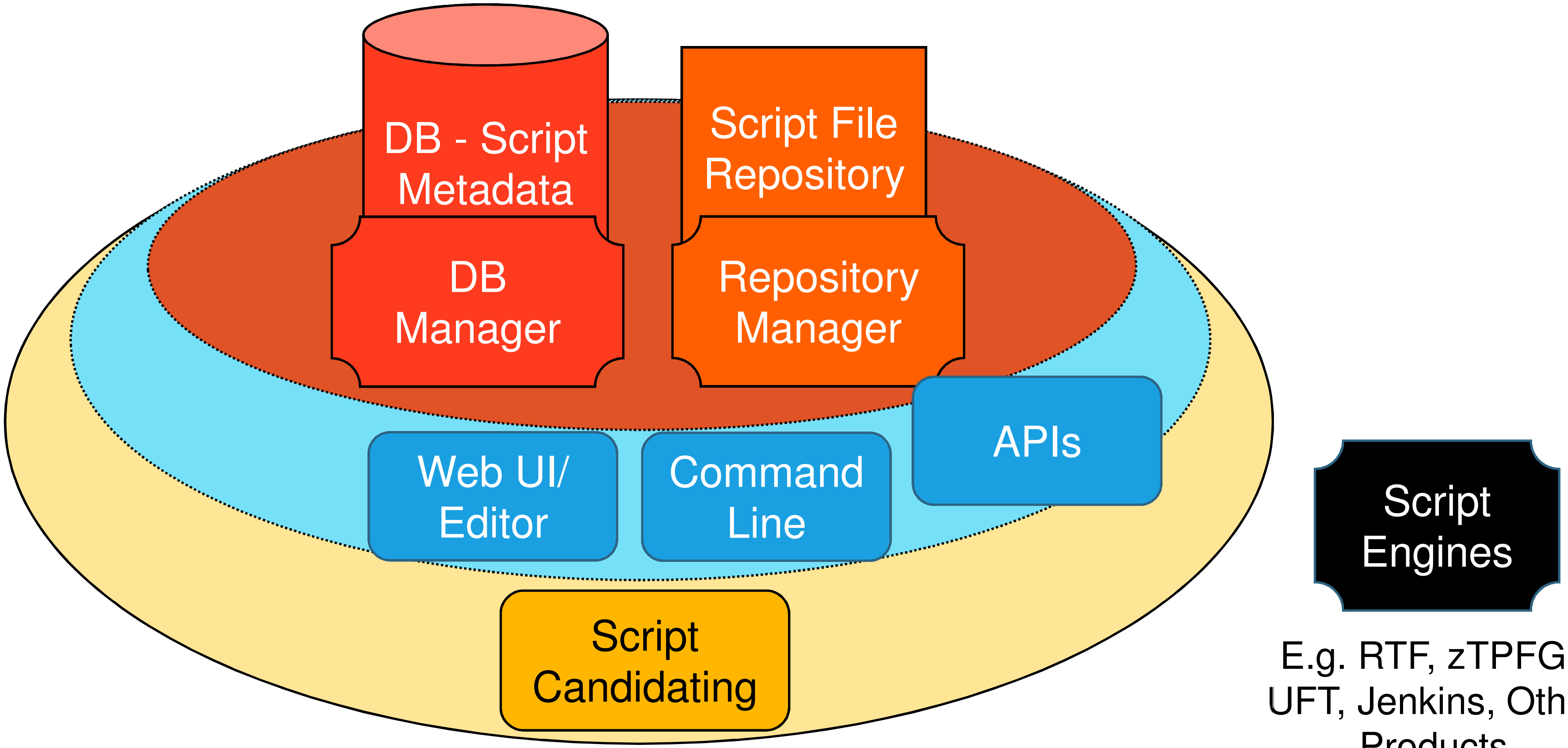
- Capture and entry
- Editing
- Storage
- Search/retrieval

…of any kind of Regression/QA/Input script, in order to produce…

- *Cost savings* due to captured knowledge and increased efficiencies in Development and Regression
- *Cost savings* through increased code base and Production quality

Scripta Architecture Overview

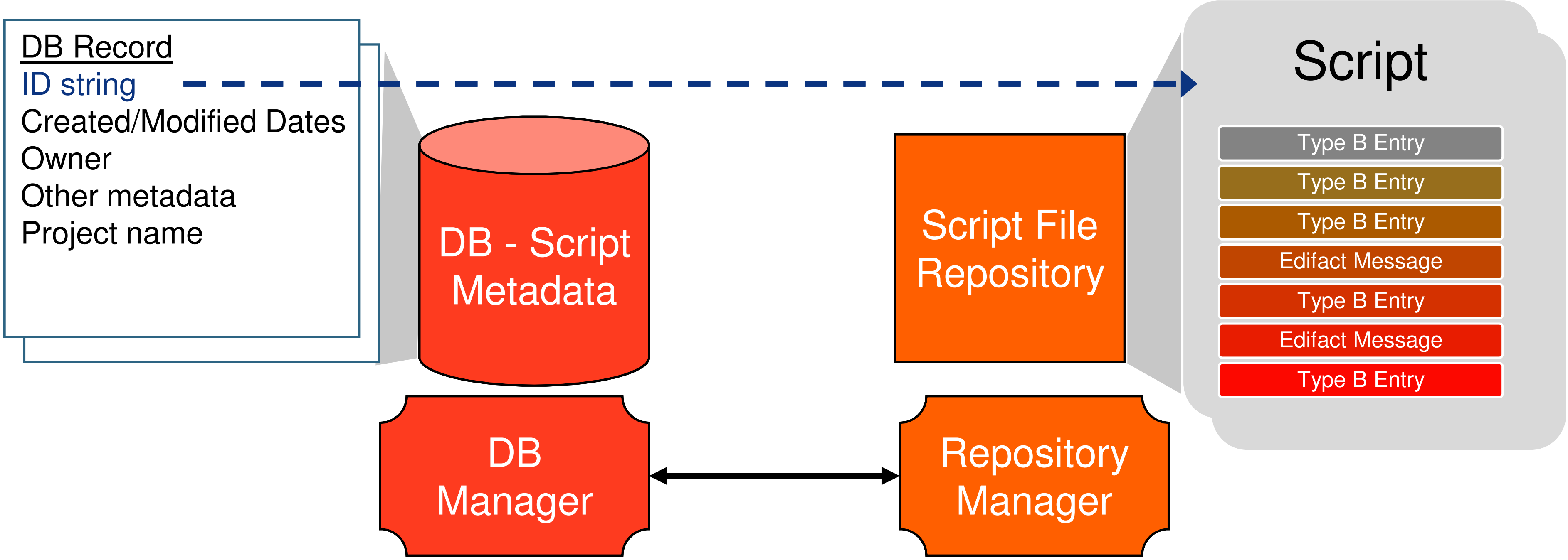
- The product includes script storage and management
- Editing, Web and command UIs, and
- Candidate script collection
- The script engines (runners) themselves lie outside the scope of the product



E.g. RTF, zTPFGI, UFT, Jenkins, Other Products

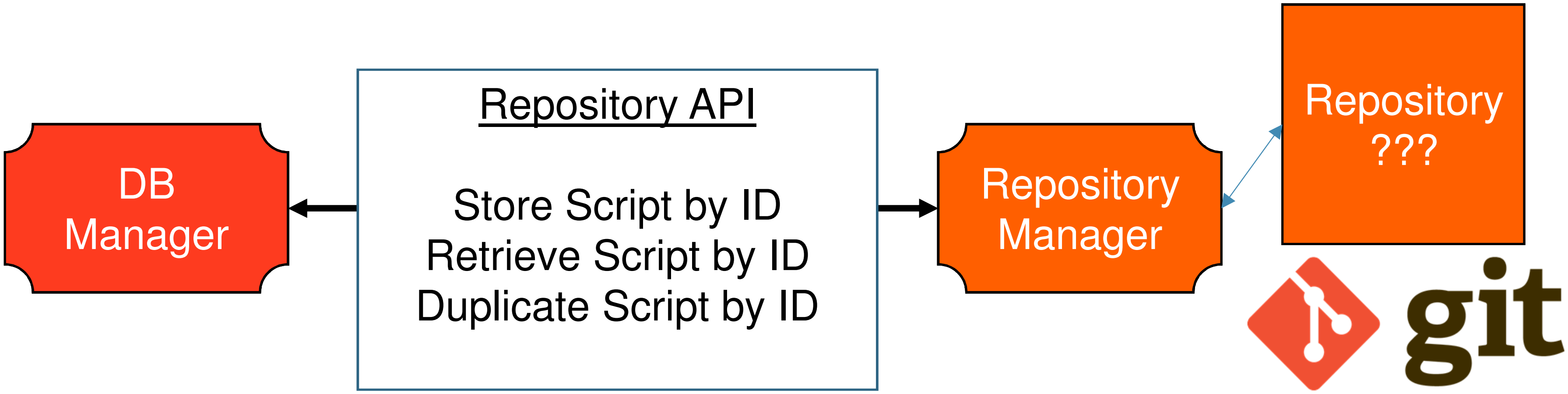
Database and Script Repository

- **Script Database** - holds DB records for each script, but not the script files themselves
- The DB record for each script contains metadata and a string to reference the script
- **Script File Repository** - holds the script files
- **Repository Manager** – performs actions on the script files, given a script reference string



Script Repository API

- The **DB Manager** does not care what technology is used for the repository
- Customers may plug in any repository technology that supports an API (user exits) and that can provide a string ID to identify the script
- TSI assumes that most customers will want to use Git for storage
- Therefore, Scripta's default repository manager will support a scheme using Git



DB Manager

Repository API

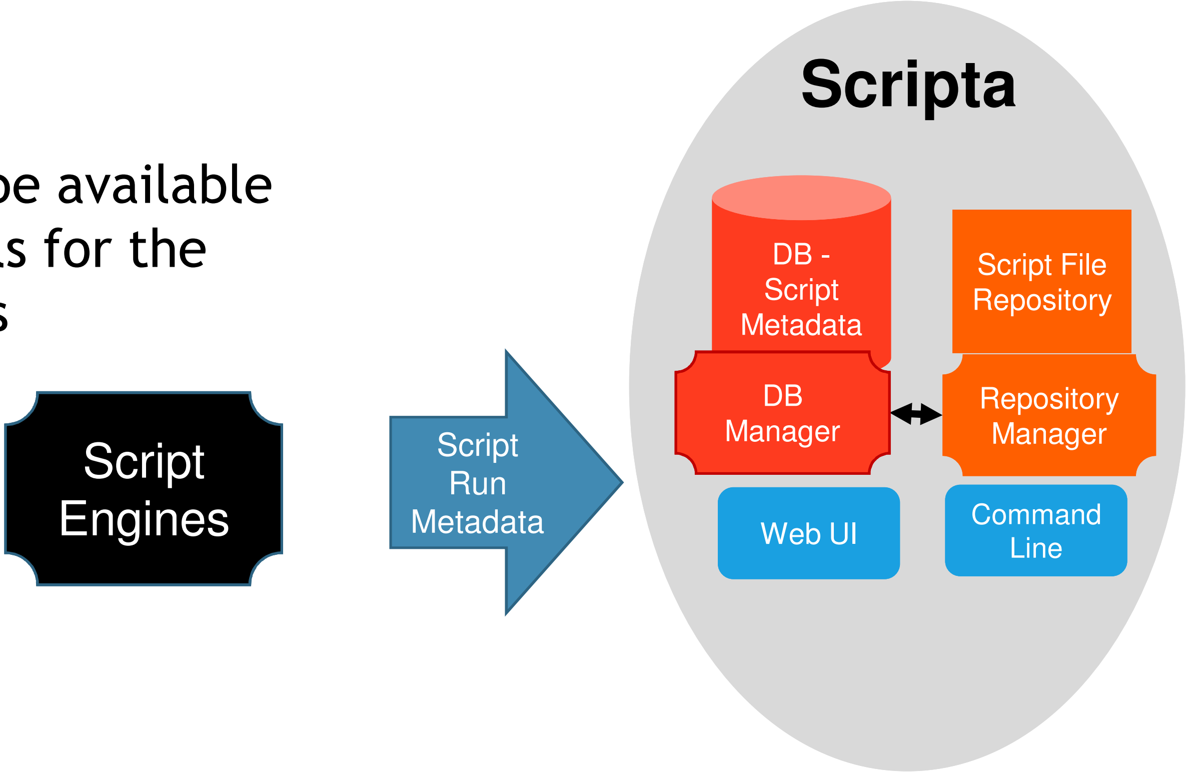
Store Script by ID
Retrieve Script by ID
Duplicate Script by ID

Repository Manager

Repository ???

Engine Metadata Into Scripta

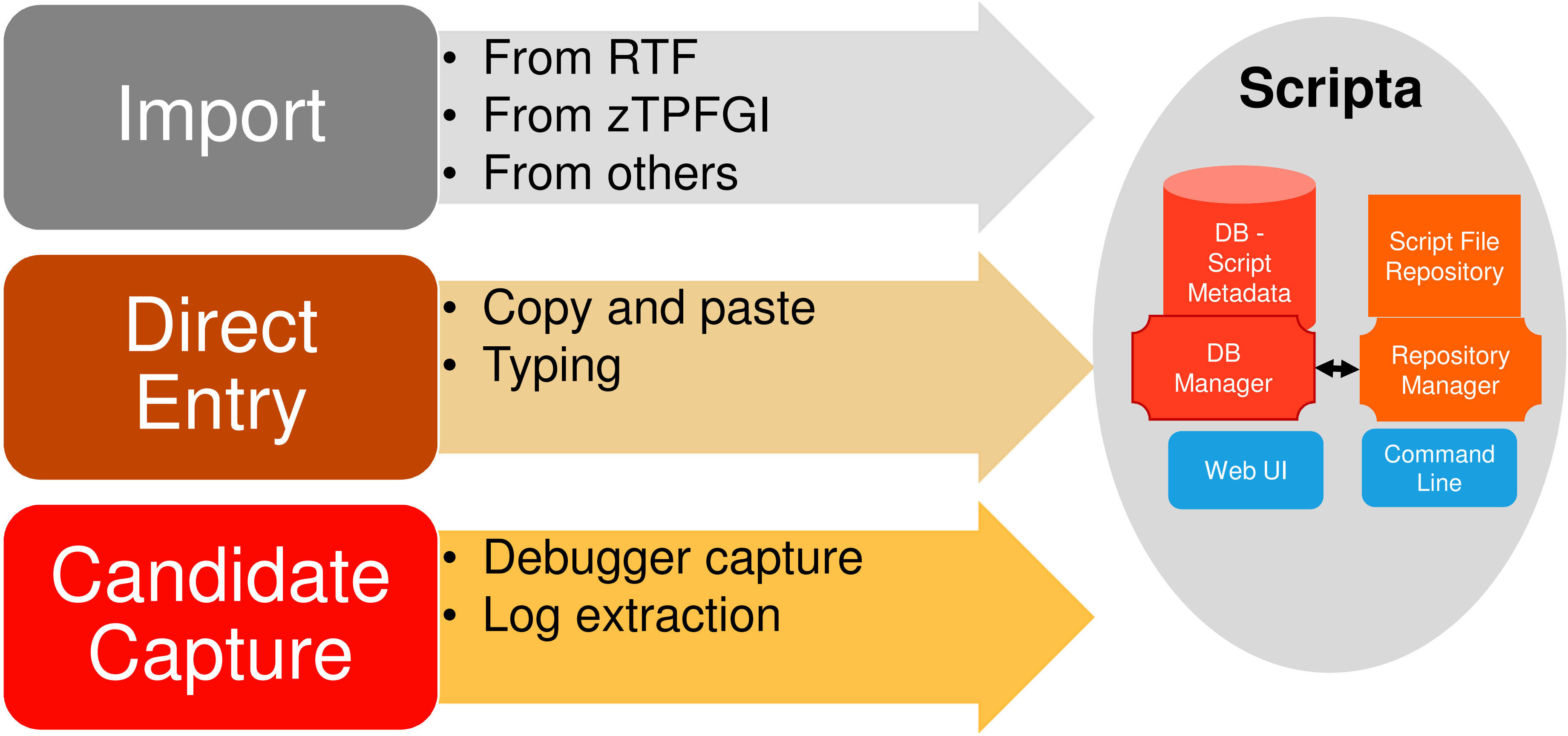
- Engines would feed data into Script about
  - Programs hit by scripts
  - Information about every run of scripts
    - Date-time
    - Type of use
    - Duration
    - Resources used
- Information would be available in the Web UI details for the scripts and searches



Script Format – Engine "Markup"

- In addition to messages/commands/entries, scripts may contain
  - Engine-specific markup
    - Notations that engine uses in processing script
    - Examples:
      - Carriage returns/continuations
      - Variables
      - Date substitutions
      - Instructions to extract parts of output and use as part of future input
    - Supported in Scripta by engine-specific plugins

Sources of Scripts in Scripta



Import

- From RTF
- From zTPFGI
- From others

Direct Entry

- Copy and paste
- Typing

Candidate Capture

- Debugger capture
- Log extraction

Scripta

DB - Script Metadata

Script File Repository

DB Manager

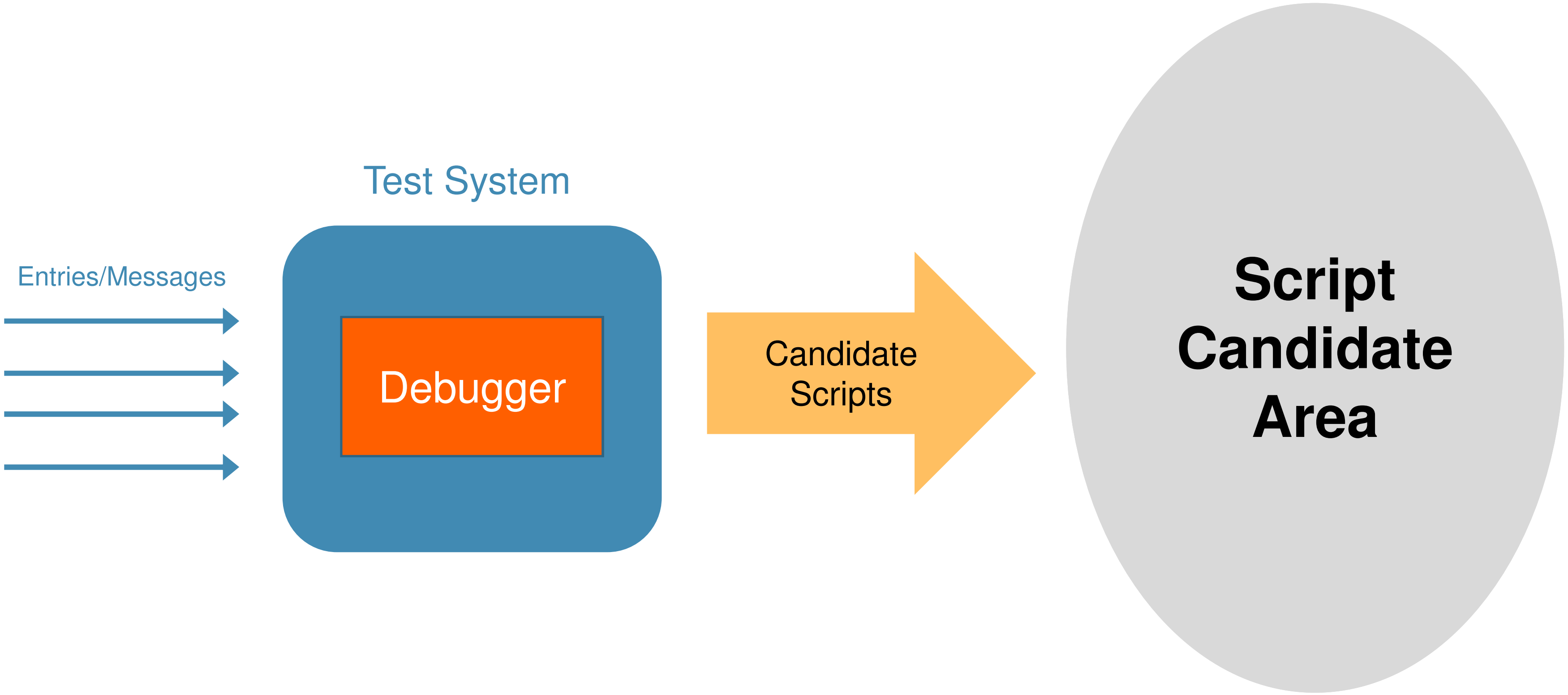
Repository Manager

Web UI

Command Line

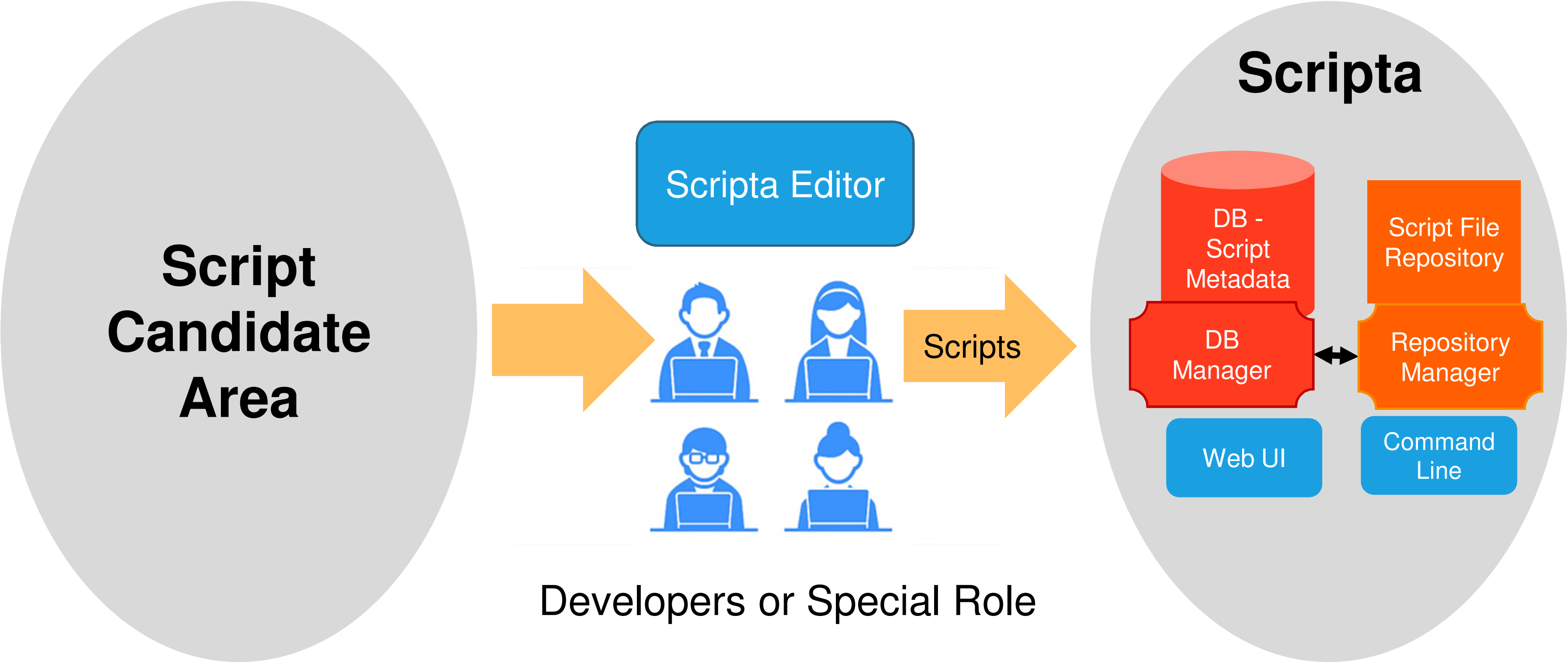
Candidate Capture

- Debugger in test system captures various types of messages
- Held for consideration in Scripta waiting area



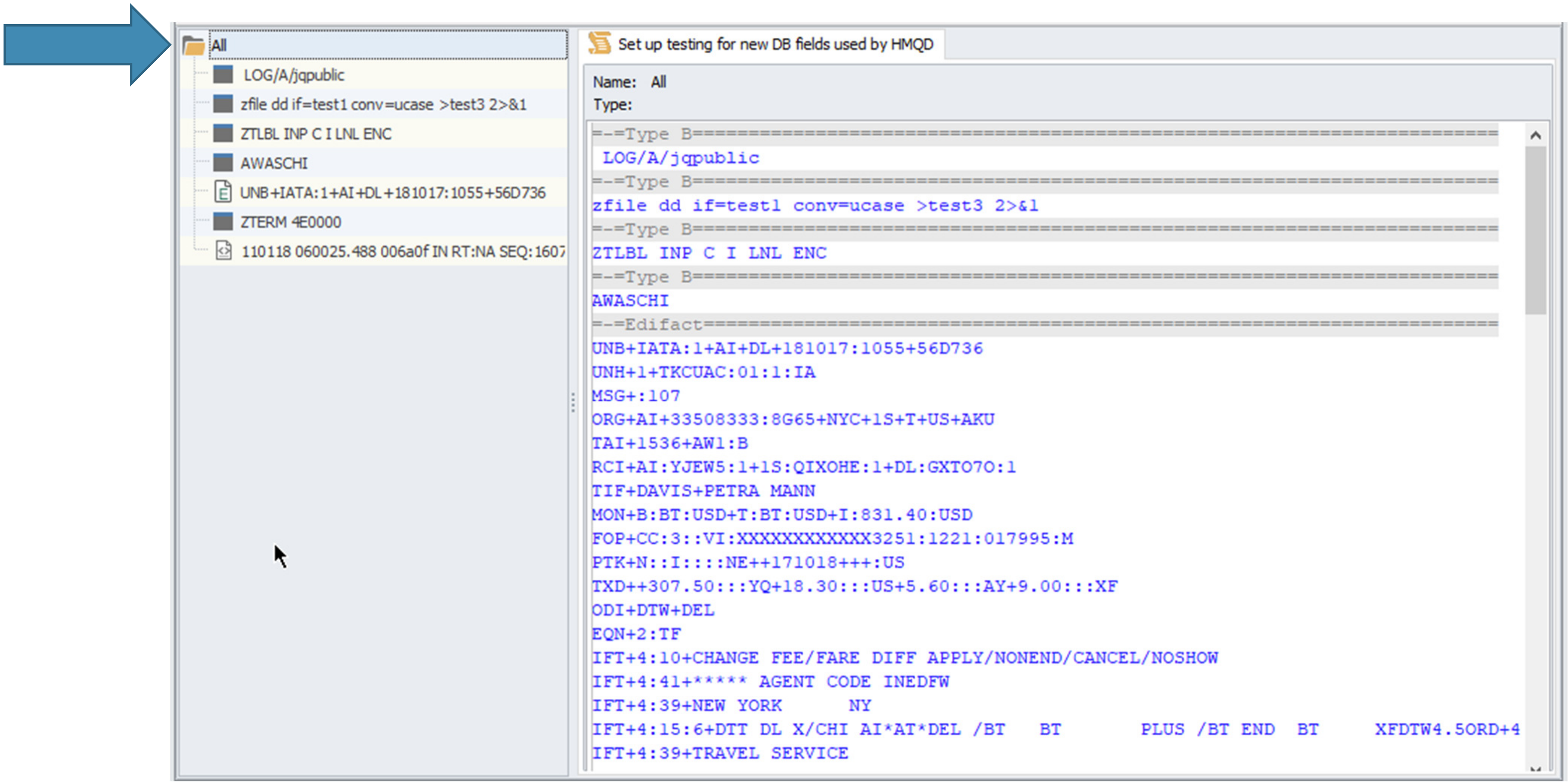
Candidate Capture – Becoming Scripts

- A developer or person in a special role periodically considers the list of saved messages/entries
- Selected messages/entries moved quickly into script(s)
- Edited in Scripta editor



Scripta Editor – Navigator & Editor

- Selecting “All” in the Navigator will display a unified view of all messages/entries in the script



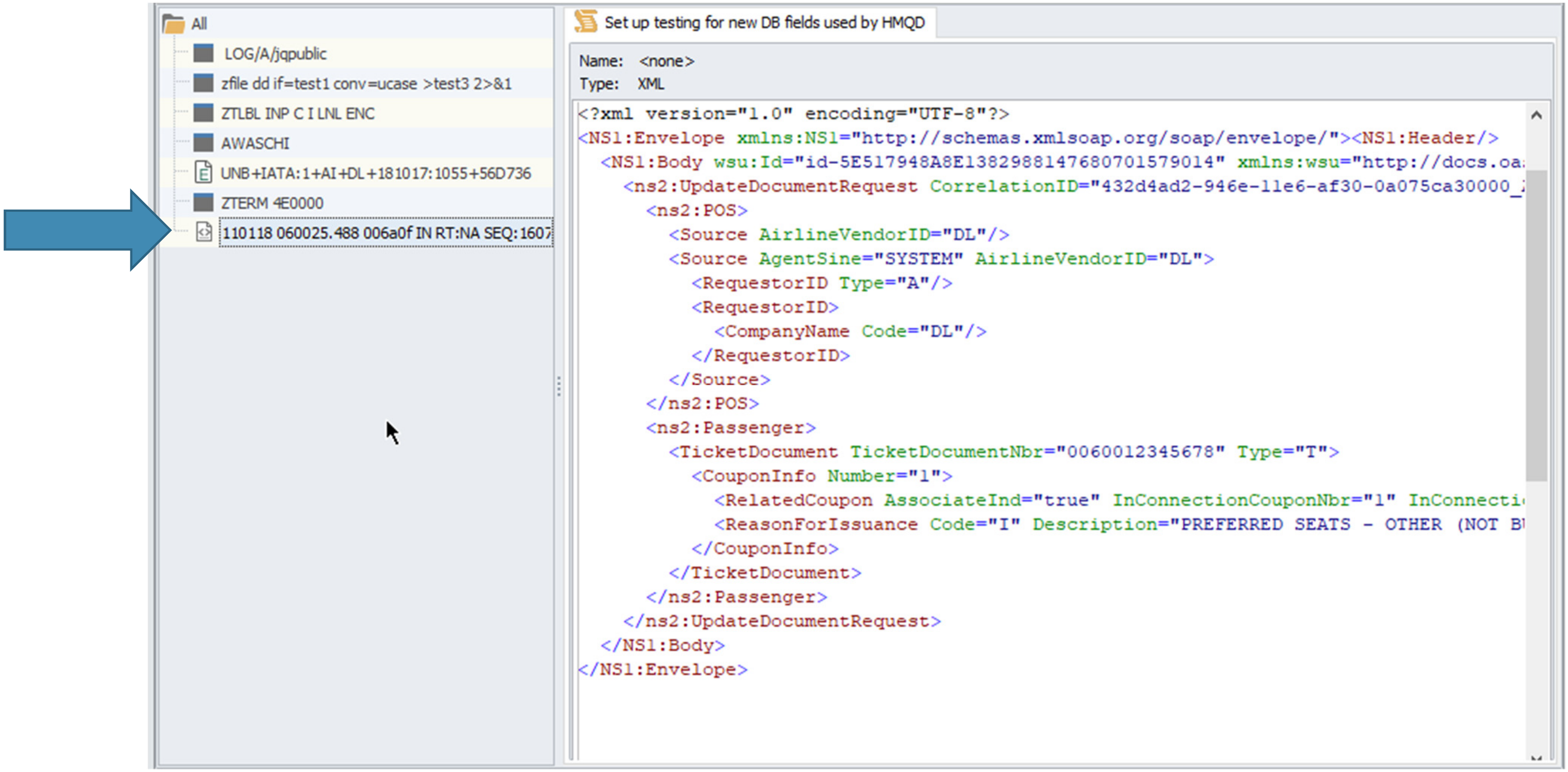
- All
  - LOG/A/jqpublic
  - zfile dd if=test1 conv=ucase >test3 2>&1
  - ZTLBL INP C I LNL ENC
  - AWASCHI
  - UNB+IATA:1+AI+DL+181017:1055+56D736
  - ZTERM 4E0000
  - 110118 060025.488 006a0f IN RT:NA SEQ:1607

Name: All
Type:

```
=-=Type B========================================================================
 LOG/A/jqpublic
=-=Type B========================================================================
zfile dd if=testl conv=ucase >test3 2>&l
=-=Type B========================================================================
ZTLBL INP C I LNL ENC
=-=Type B========================================================================
AWASCHI
=-=Edifact=======================================================================
UNB+IATA:1+AI+DL+181017:1055+56D736
UNH+1+TKCUAC:01:1:IA
MSG+:107
ORG+AI+33508333:8G65+NYC+1S+T+US+AKU
TAI+1536+AW1:B
RCI+AI:YJEW5:1+1S:QIXOHE:1+DL:GXTO7O:1
TIF+DAVIS+PETRA MANN
MON+B:BT:USD+T:BT:USD+I:831.40:USD
FOP+CC:3::VI:XXXXXXXXXXXX3251:1221:017995:M
PTK+N::I::::NE++171018+++:US
TXD++307.50:::YQ+18.30:::US+5.60:::AY+9.00:::XF
ODI+DTW+DEL
EQN+2:TF
IFT+4:10+CHANGE FEE/FARE DIFF APPLY/NONEND/CANCEL/NOSHOW
IFT+4:41+***** AGENT CODE INEDFW
IFT+4:39+NEW YORK      NY
IFT+4:15:6+DTT DL X/CHI AI*AT*DEL /BT   BT       PLUS /BT END  BT     XFDTW4.5ORD+4
IFT+4:39+TRAVEL SERVICE
```

Scripta Editor – Navigator & Editor

- Selecting a specific message in the Navigator will display that message alone



LOG/A/jqpublic

zfile dd if=test1 conv=ucase >test3 2>&1

ZTLBL INP C I LNL ENC

AWASCHI

UNB+IATA:1+AI+DL+181017:1055+56D736

ZTERM 4E0000

110118 060025.488 006a0f IN RT:NA SEQ:1607

Set up testing for new DB fields used by HMQD

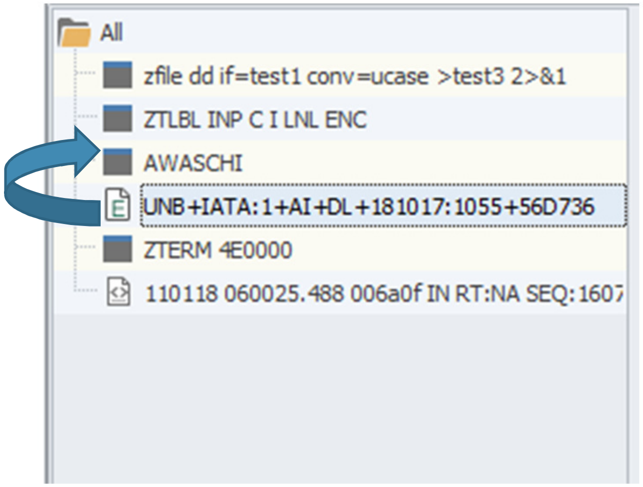
Name: <none>

Type: XML

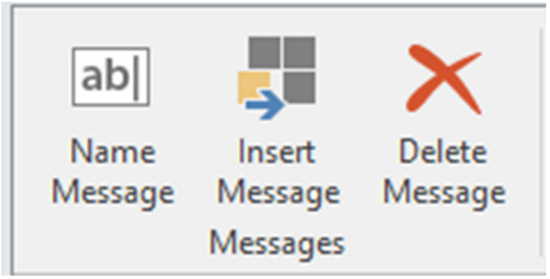
```
<?xml version="1.0" encoding="UTF-8"?>
<NS1:Envelope xmlns:NS1="http://schemas.xmlsoap.org/soap/envelope/"><NS1:Header/>
  <NS1:Body wsu:Id="id-5E517948A8E1382988147680701579014" xmlns:wsu="http://docs.oa
    <ns2:UpdateDocumentRequest CorrelationID="432d4ad2-946e-11e6-af30-0a075ca30000_
      <ns2:POS>
        <Source AirlineVendorID="DL"/>
        <Source AgentSine="SYSTEM" AirlineVendorID="DL">
          <RequestorID Type="A"/>
          <RequestorID>
            <CompanyName Code="DL"/>
          </RequestorID>
        </Source>
      </ns2:POS>
      <ns2:Passenger>
        <TicketDocument TicketDocumentNbr="0060012345678" Type="T">
          <CouponInfo Number="1">
            <RelatedCoupon AssociateInd="true" InConnectionCouponNbr="1" InConnecti
            <ReasonForIssuance Code="I" Description="PREFERRED SEATS - OTHER (NOT B
          </CouponInfo>
        </TicketDocument>
      </ns2:Passenger>
    </ns2:UpdateDocumentRequest>
  </NS1:Body>
</NS1:Envelope>
```

Navigator – Other Features

Navigator



```
All
zfile dd if=test1 conv=ucase >test3 2>&1
ZTLBL INP C I LNL ENC
AWASCHI
UNB+IATA:1+AI+DL+181017:1055+56D736
ZTERM 4E0000
110118 060025.488 006a0f IN RT:NA SEQ:1607
```



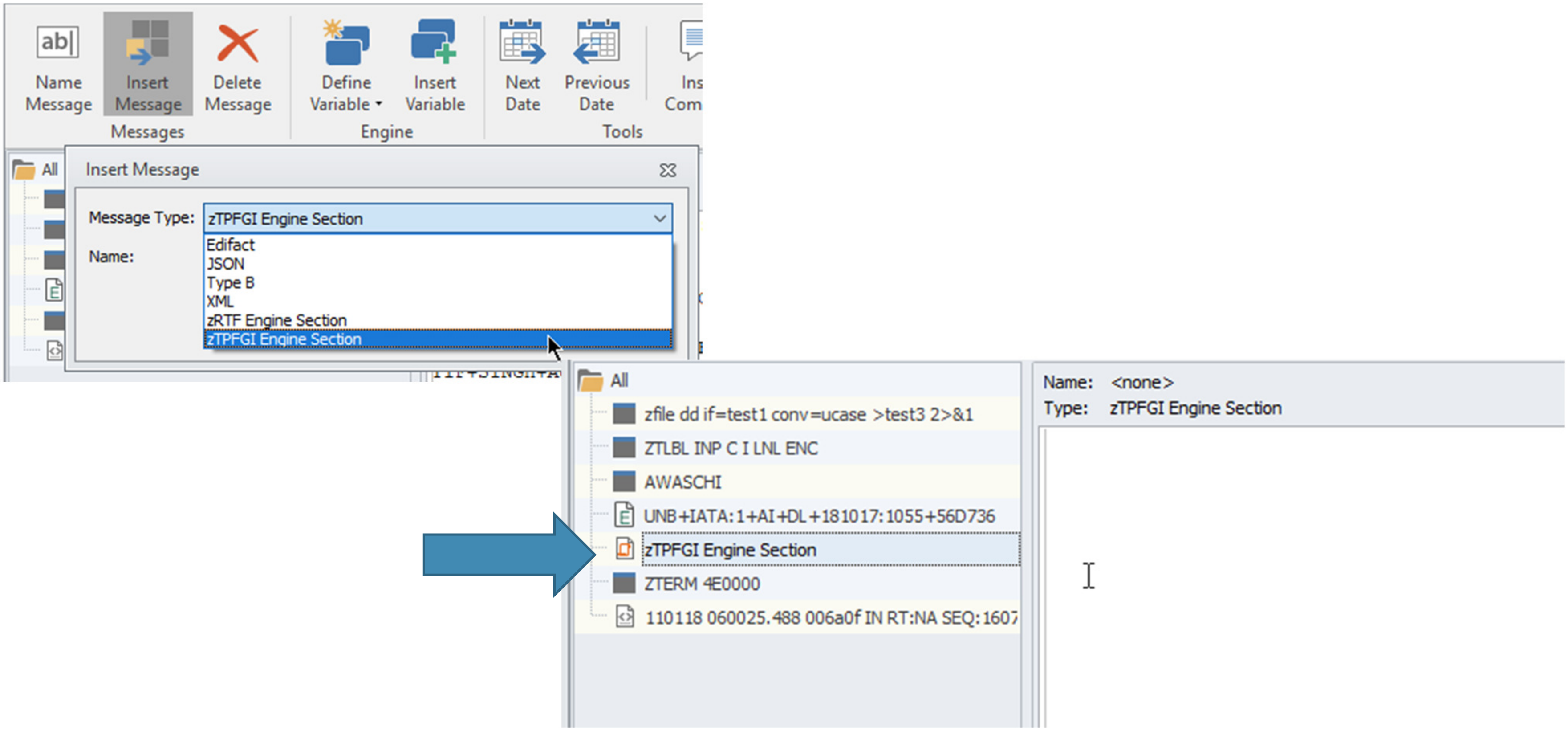
- Messages/Entries can be rearranged in the script by dragging and dropping items in the Navigator
- Messages/Entries can be named/described and that text will appear in the Navigator
  - E.g. “Reserves Seat”
- Insert space for a new message/entry and paste text into the editor area for it (or type)
- Auto-recognition of message/entry types (via extensible plug-in system)
- “Include” utility scripts

What is a “Utility Script”?

- A script containing a sequence of message/entries that are frequently required by other scripts
  - Login
  - Get a PNR for rest of script to use
- Hardcoding the messages/entries would make the system less flexible
- Tell Scripta you want to include the utility script
  - It will embed a reference to the script
  - It will incorporate the utility script before giving to engine

Navigator/Editor – Engine Specific Instructions

- Script editor can be used to insert sections of instructions to engines
- Engines can use these sections to define variables and process output



Message Type: zTPFGI Engine Section

Name:

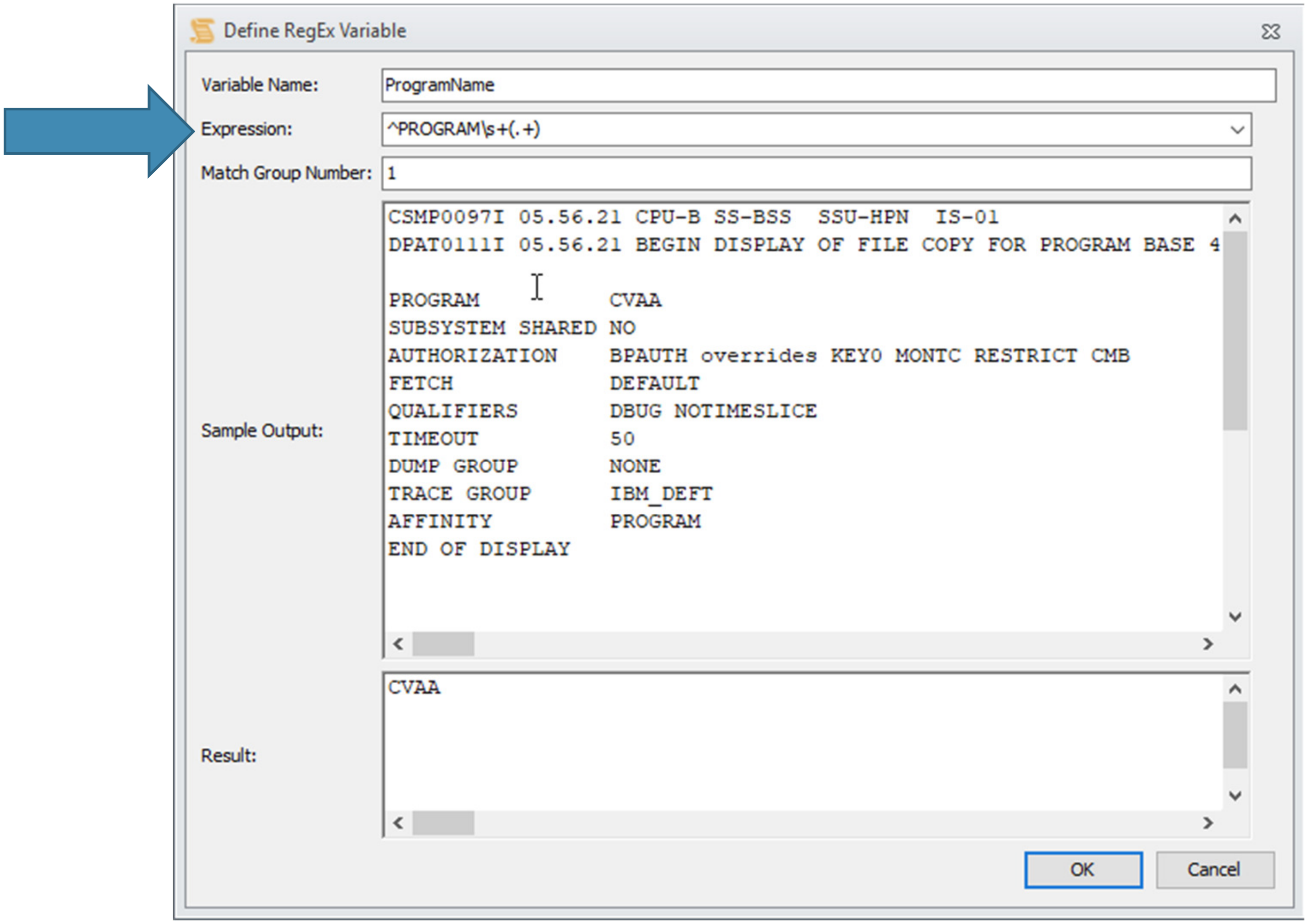
Edifact
JSON
Type B
XML
zRTF Engine Section
zTPFGI Engine Section

All
- zfile dd if=test1 conv=ucase >test3 2>&1
- ZTLBL INP C I LNL ENC
- AWASCHI
- UNB+IATA:1+AI+DL+181017:1055+56D736
- zTPFGI Engine Section
- ZTERM 4E0000
- 110118 060025.488 006a0f IN RT:NA SEQ:1607

Name: <none>
Type: zTPFGI Engine Section

Editor – RegEx Variables

- For processing output (for use in future input), info can be extracted into variables using regular expressions
- A wizard lets you develop the expression based on a sample output



Variable Name: ProgramName

Expression: ^PROGRAM\s+(.+)

Match Group Number: 1

Sample Output:

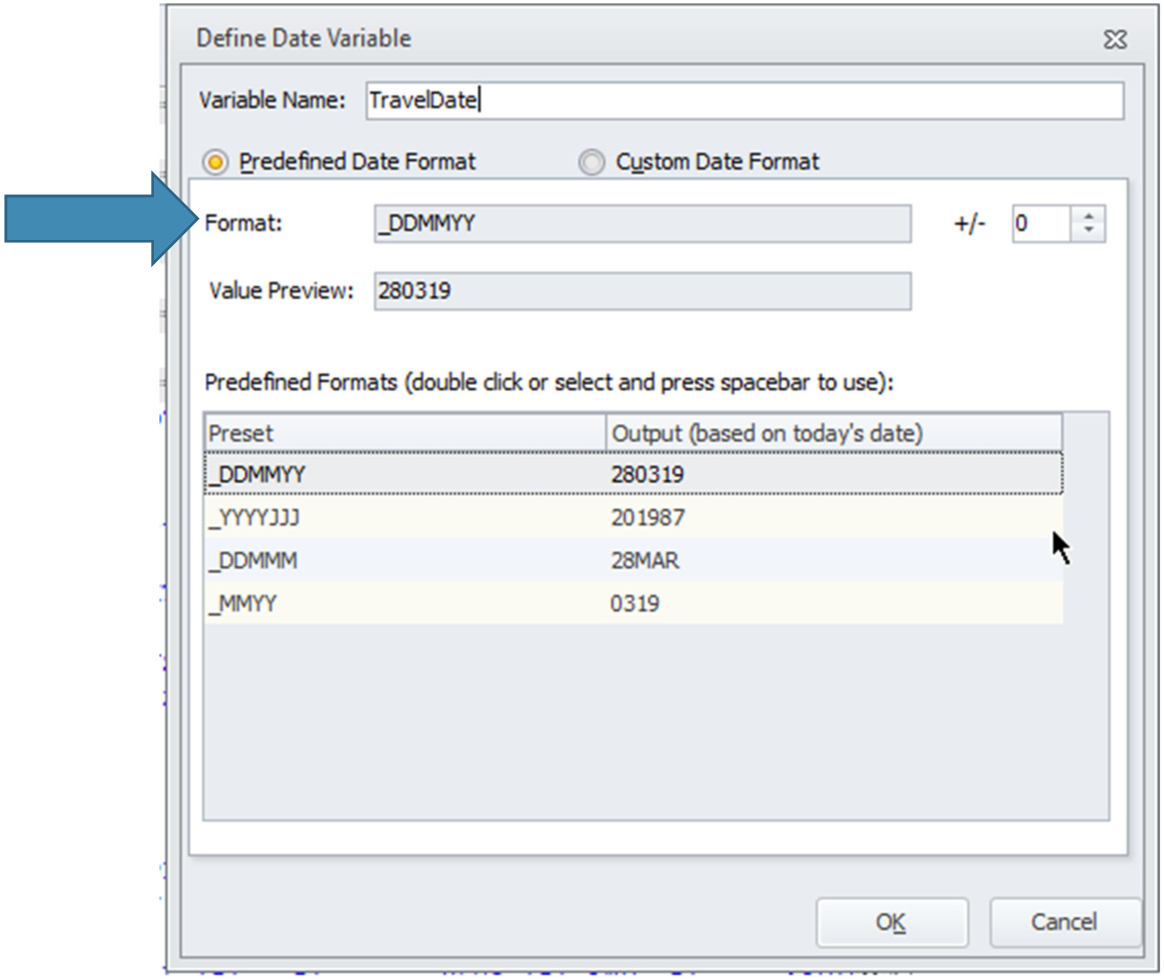
```
CSMP0097I 05.56.21 CPU-B SS-BSS  SSU-HPN  IS-01
DPAT0111I 05.56.21 BEGIN DISPLAY OF FILE COPY FOR PROGRAM BASE 4

PROGRAM          CVAA
SUBSYSTEM SHARED NO
AUTHORIZATION    BPAUTH overrides KEY0 MONTC RESTRICT CMB
FETCH            DEFAULT
QUALIFIERS       DBUG NOTIMESLICE
TIMEOUT          50
DUMP GROUP       NONE
TRACE GROUP      IBM_DEFT
AFFINITY         PROGRAM
END OF DISPLAY
```

Result:

```
CVAA
```

Editor – Date Variables

- Wizards allow users to develop date expressions that replace actual dates in the inputs



Define Date Variable

Variable Name: TravelDate

● Predefined Date Format ○ Custom Date Format

Format: _DDMMYY +/- 0

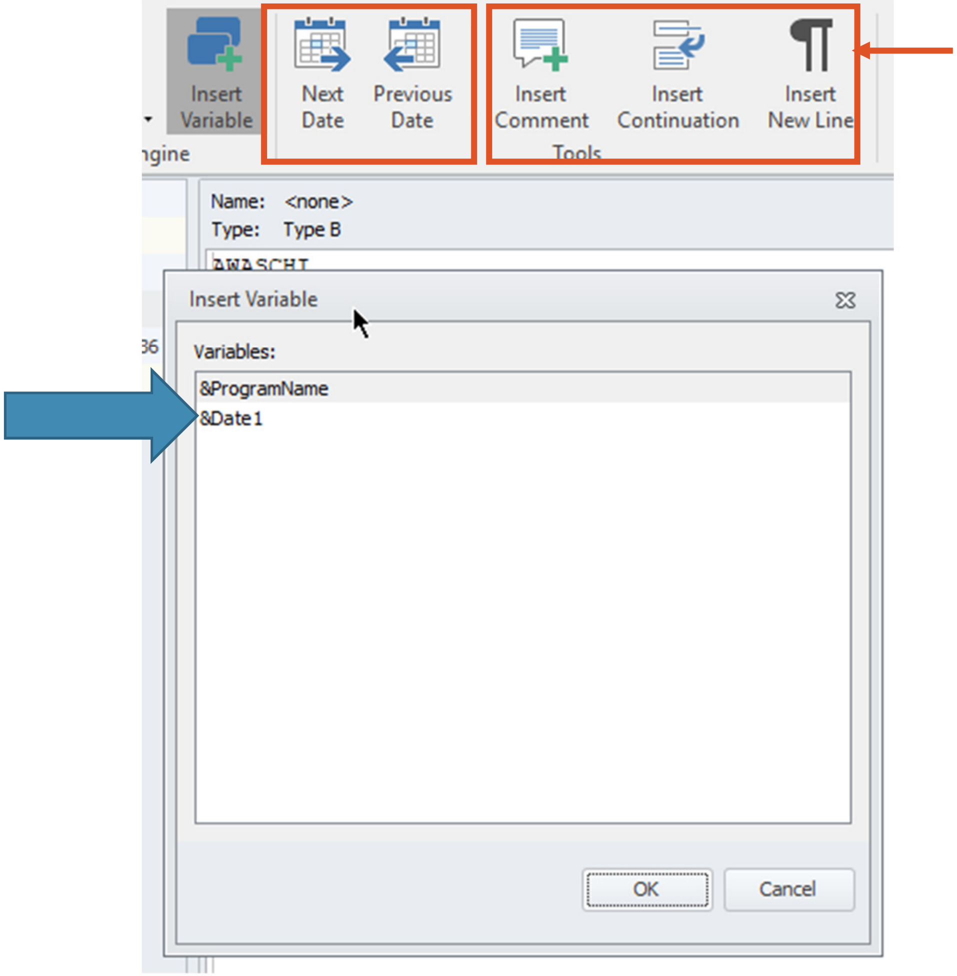
Value Preview: 280319

Predefined Formats (double click or select and press spacebar to use):

Preset	Output (based on today's date)
_DDMMYY	280319
_YYYYJJJ	201987
_DDMMM	28MAR
_MMYY	0319

Editor – Dates & Other Editing Tools

- Let Scripta find and highlight likely dates that need substitution
- Then use the variable tools to create and insert substitute variables



Name: <none>
Type: Type B

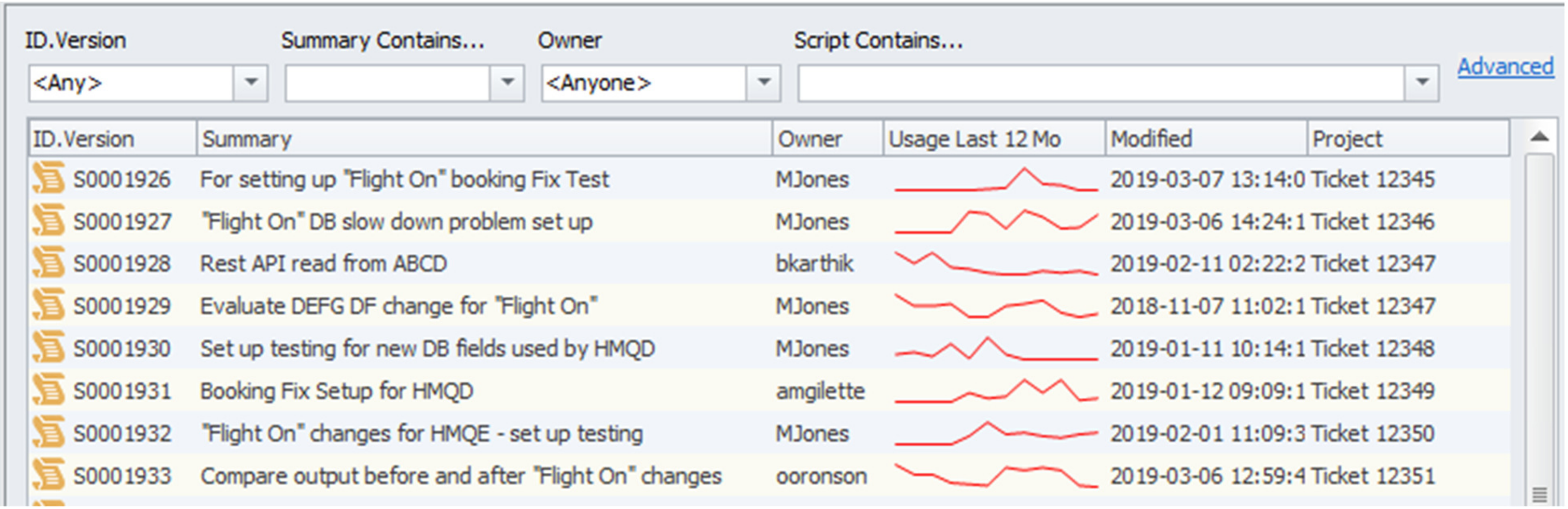
AWASCHI

Variables:
&ProgramName
&Date1

- Insert comments and other formatting understood by the engines as well

User Script Database Search

- The Web UI also allows developers and zRTF and other staff to search for scripts by…
  - ID
  - Summary
  - Owner
  - Script text
- Advanced options include other metadata and fields
  - Programs hit by script supported but info must be supplied by engines

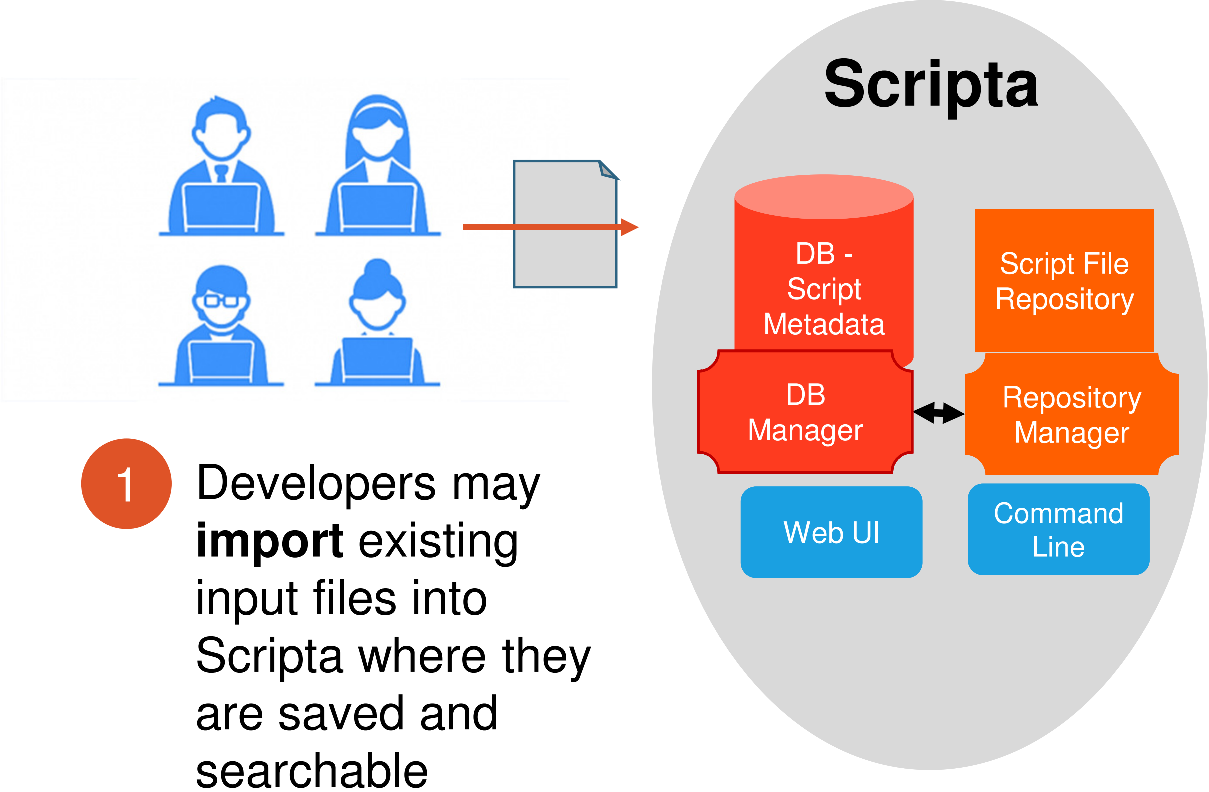


ID.Version	Summary Contains…	Owner	Script Contains…
<Any>		<Anyone>	

ID.Version	Summary	Owner	Usage Last 12 Mo	Modified	Project
S0001926	For setting up "Flight On" booking Fix Test	MJones		2019-03-07 13:14:0	Ticket 12345
S0001927	"Flight On" DB slow down problem set up	MJones		2019-03-06 14:24:1	Ticket 12346
S0001928	Rest API read from ABCD	bkarthik		2019-02-11 02:22:2	Ticket 12347
S0001929	Evaluate DEFG DF change for "Flight On"	MJones		2018-11-07 11:02:1	Ticket 12347
S0001930	Set up testing for new DB fields used by HMQD	MJones		2019-01-11 10:14:1	Ticket 12348
S0001931	Booking Fix Setup for HMQD	amgilette		2019-01-12 09:09:1	Ticket 12349
S0001932	"Flight On" changes for HMQE - set up testing	MJones		2019-02-01 11:09:3	Ticket 12350
S0001933	Compare output before and after "Flight On" changes	ooronson		2019-03-06 12:59:4	Ticket 12351

Developer Workflow – Editing/Input

- Scripta does not require zTPFGI, but can be embedded in zTPFGI for developer use



1 Developers may **import** existing input files into Scripta where they are saved and searchable

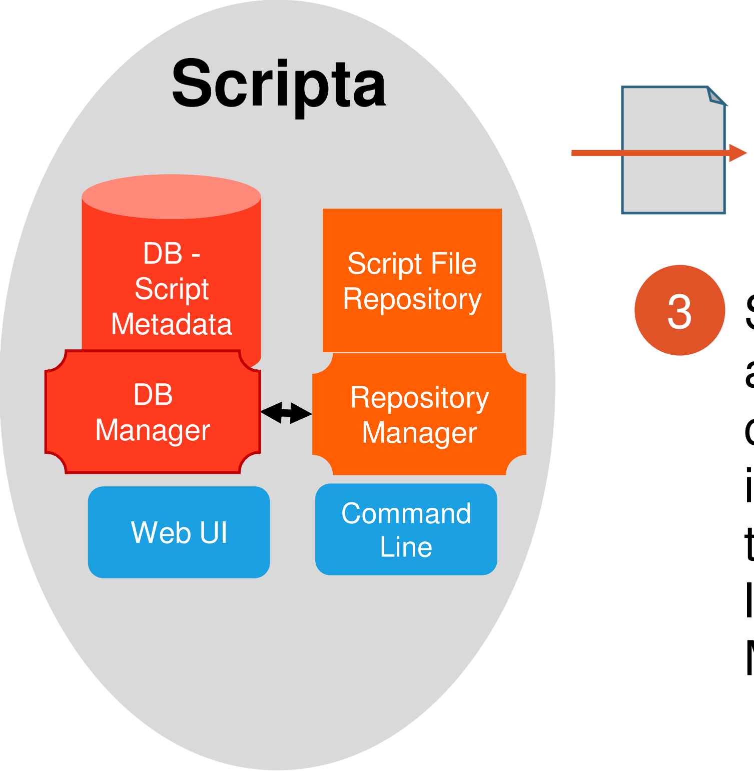
1a Developers may also develop new input files directly in Scripta, making use of its advanced editing features

1c Developers can also discover and create scripts by selecting commands/entries/ messages from the Candidate script area of Scripta

Developer Workflow – Search/Selection/ Execution

- When developers need a script, they can go to Scripta rather than their own ad hoc file system

2 Developers find their script using Scripta, double click



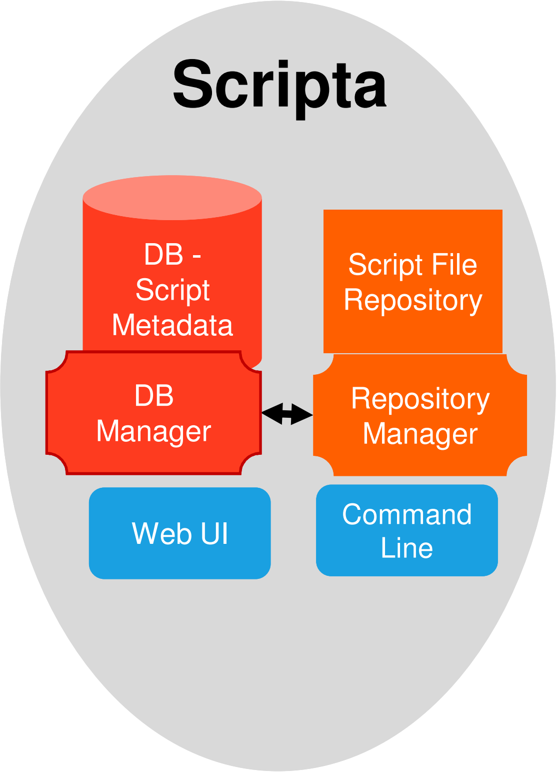
3 Scripta automatically converts to zTPFGI input file format and the input file is loaded into zTPFGI Message Run

- Scripta storage and search allows easier sharing of scripts between team members as well

Regression Workflow – New Script Sources

- Scripta provides Regression with new script sources in a couple of ways

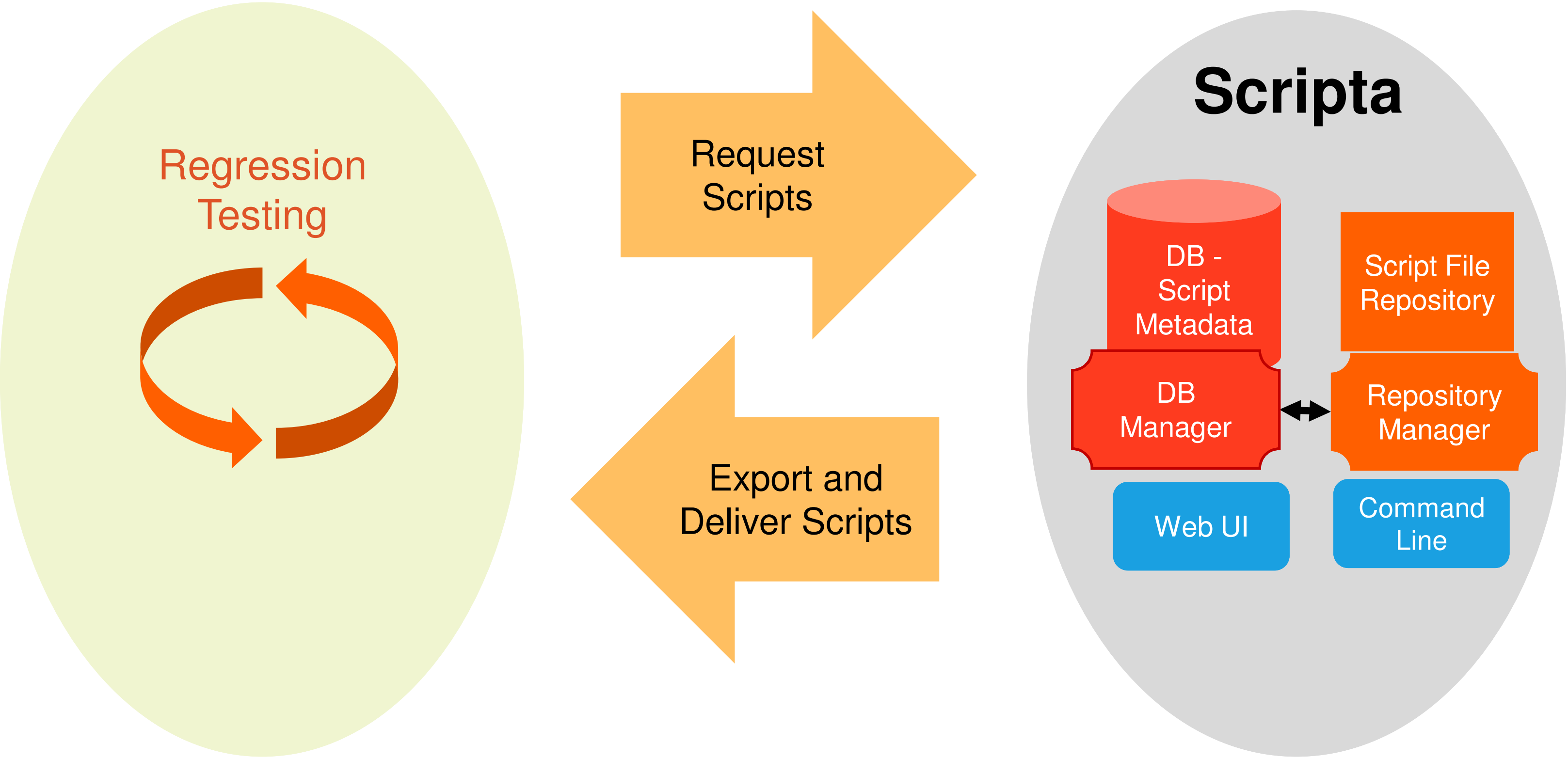
1 Developer scripts in Scripta are a great source of new scripts to export for regression



1a The Scripta Candidate area (scripts collected from the test systems by the debugger) can also be a treasury of new scripts

Regression Workflow – Storage & Use

- Regression scripts can be maintained in Scripta and automatically exported for regression runs via the Scripta API



- This allows Regression to take advantage of Scripta editing and search features

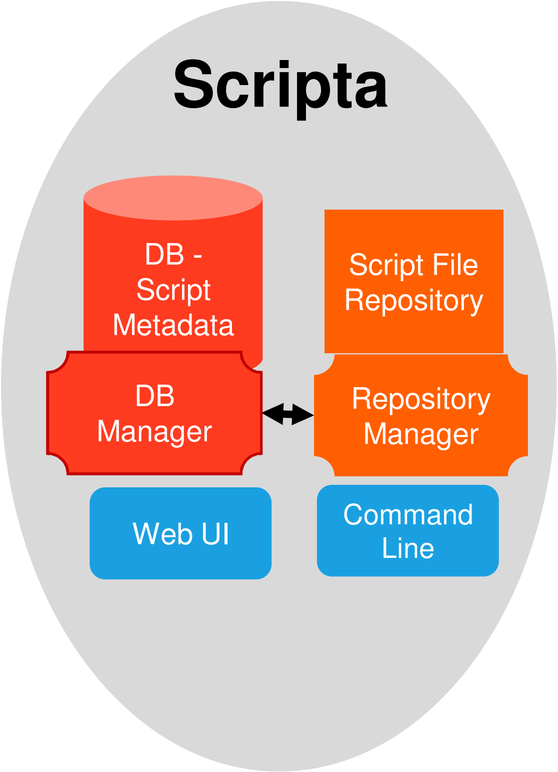
Benefits

- Capturing existing organizational knowledge and resources
  - Capture of script knowledge that may be lost now
  - Conversion of script knowledge into scripts
- Increase in quality of codebase, less chance of problems in Production
  - Increased use of scripts by developers
  - More comprehensive scripts available for testing/Regression
- Increased efficiency in set up of scripts and testing/debugging
  - Easier, more efficient test/debug set up in Development
  - More efficient creation and editing of scripts by developers and Regression

Benefits of Scripta Over Workflow Tools Alone

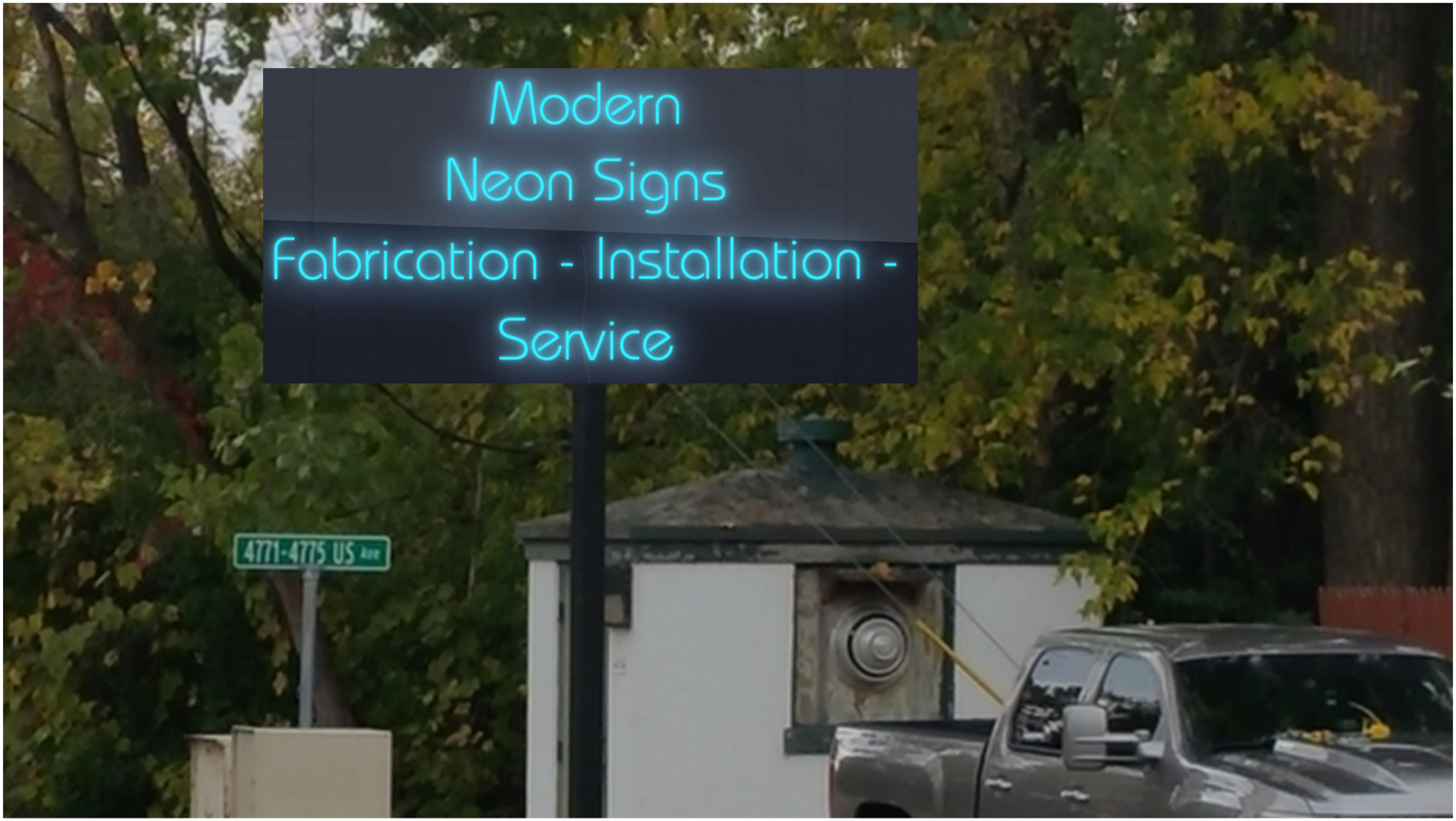
- Search tools
  - Script synergy between Development and Regression
- Special supported features, such as utility scripts
- Advanced editing support

Conclusion



- Scripta being developed for release in Fall 2019
- Would you like to influence the features in Scripta to be best for your company? Contact us

Goal: Opportunities Capitalized Upon



Questions?

TSI

MOVE FORWARD ▸ TOGETHER

Thank you