

Single Day Event

AI-Driven Validation of Edit Checks in EDC Based on the User Requirement Specification (URS)

Flora Chen
Novartis



Disclaimer

The views and opinions expressed in this presentation are those of the author and do not necessarily reflect the official policy or position of Novartis or any of its officers.



Example of Edit Check

Query Field	Validation Name	User Requirement	Edit Check Message
If (AGE_DM001 in Demographics with record position 0 IsLessThan 18) And (AGE_DM001 in Demographics with record position 0 IsNotEmpty) then... open a query to Site from System on AGE_DM001 in Demographics with record position 0, displaying "Age should be greater or equal 18 years. Please verify and update as appropriate, else clarify."(requires response)(requires manual close)			

Check Steps

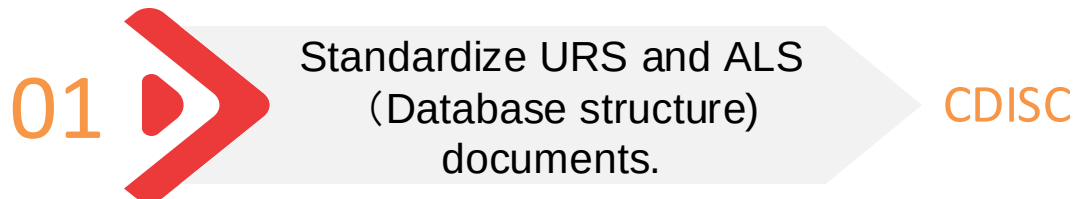
	Type	Step	Edit
	Data Value	...>Demographics>AGE_DM001>AGE_DM001>0>...>...>None	
	Constant	18 (3)	
	Check Function	IsLessThan	
	Data Value	...>Demographics>AGE_DM001>AGE_DM001>0>...>...>None	
	Check Function	IsNotEmpty	
	Check Function	And	
Add Check Step			

Check Actions

Data Point	Action	Edit
...>Demographics>AGE_DM001>AGE_DM001>0>...>...>None	Open Query: Site from System: Age should be greater or equal 18 years. Please verify and update as appropriate, else clarify. (requires response) (requires manual close)	

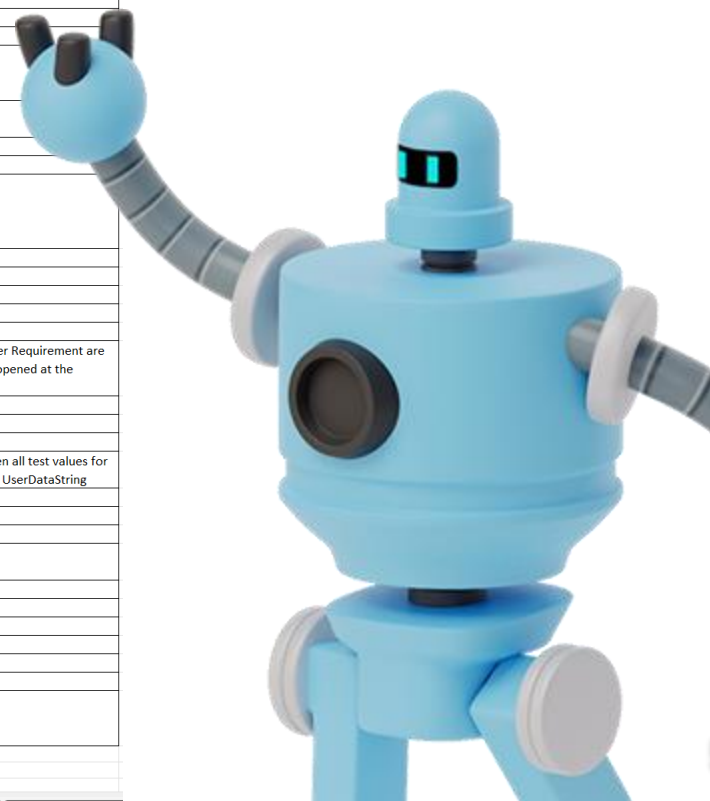


AI-Driven URS Automation Process



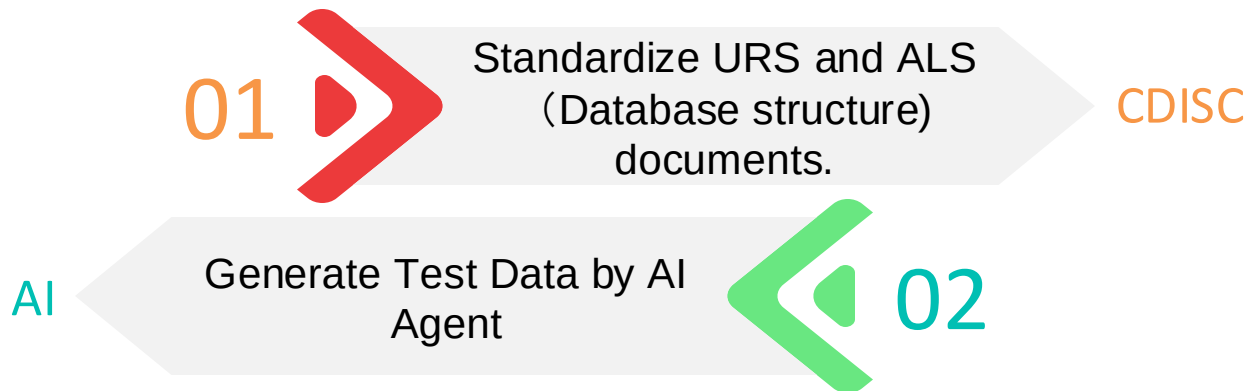
Essential Elements

SheetName	ColumnName	Meaning	Special instruction
Edit Checks	Form OID	OID of Form	
Edit Checks	Form Name	Name of form	
Edit Checks	Edit check category	EC classification, related to query is Edit Check, and related to dynamic and value derivation are Dynamic and Derivation	
Edit Checks	Query Field	The data point where Query is located after it meets the conditions and is triggered	
Edit Checks	Validation Name	Edit Check Name	
Edit Checks	User Requirement	Logic	
Edit Checks	Special Instructions(For example exclusion of certain folders, partial dates, study Arms etc..)		
Edit Checks	Check Action	Set Data Point	
Edit Checks	Check Action	Derivation	
Edit Checks	Check Action	Open Query	
Edit Checks	Check Action	Set Datapoint Visible	
Edit Checks	Check Action	Derive Field	
Edit Checks	Edit Check Message	Query text	If the conditions that trigger the query in the User Requirement are met, a query with the following content will be opened at the corresponding data point
Fields	FormOID	OID of Form	
Fields	FieldOID	OID of Field	Connected with Query Field
Fields	DataFormat	Format of data	
Fields	DataDictionaryName	Field value codelist	If this data point has a DataDictionaryName, then all test values for this data point should be randomly picked in the UserDataString
Fields	PreText	Natural text corresponding to data points	User Requirement in Edit Checks contains this
Fields	IsLog	Whether the datapoint is a logline data	
Fields	DefaultValue	Default Value of this datapoint	
Fields	IsVisible	In the initial state, whether the data points are visible	
Folders	FolderOID	OID of folder	
Folders	FolderName	Name of folder	
Folders	IsReusable	Will the folder appear repeatedly?	
DataDictionaryEntries	DataDictionaryName	Field value codelist	Same as the datadictionary in fields
DataDictionaryEntries	CodedData	System coded data in the backend	
DataDictionaryEntries	UserDataString	User entry data	Connected with Codedata
DataDictionaryEntries	Specify	The Yes or No of this answer indicates whether this dictionary requires additional input in addition to the option itself.	





AI-Driven URS Automation Process





Generate Test Plan by AI Agent

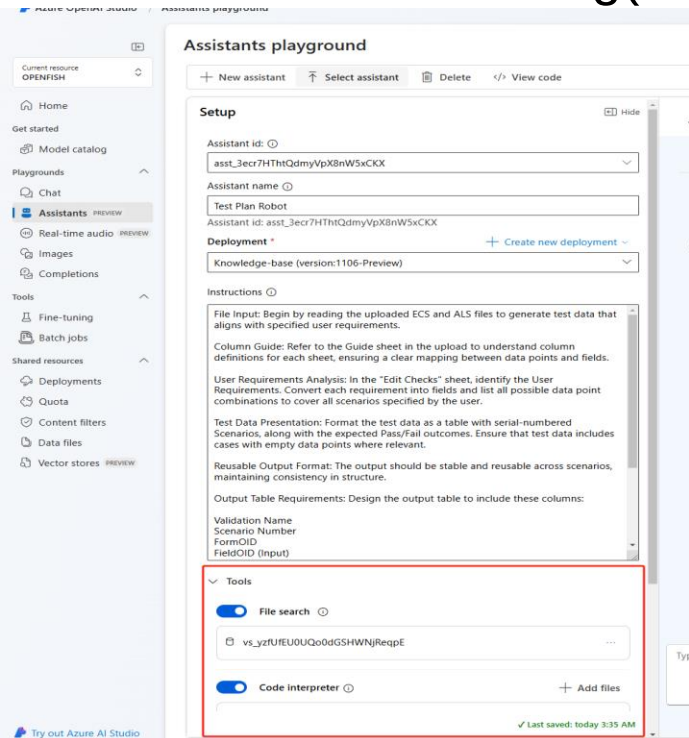
- Extract necessary information
- Combine the information with ALS (Folder +Form+Field+Logline+ Dataformat+ DataDictionaryEntries)
- Upload the related files to AI Agent
 - Prompts
 - Output format config
 - Adjust the prompts with outputs





Generate Test Plan by AI Agent

■ AI Service Config(Azure)



Classic ECS and Test Plan Configuration



**If Outcome = Not recovered/not resolved OR Unknown AND
End Date <> empty THEN fire query**

"If Outcome = Not recovered/not resolved OR Unknown AND End Date <> empty THEN fire query."

To convert this into field-level information for generating test data, we can identify the following fields:

Field Information:

1. FormOID: AEG001
2. FieldOID (Input):
 - AEOUT (Outcome)
 - AEENDTC (End Date)
3. Conditions:
 - AEOUT (Outcome) should be either "Not recovered/not resolved" or "Unknown".
 - AEENDTC (End Date) should not be empty.
4. Expected Action:
 - A query should be fired when the conditions are met.
 - The Query Field will likely refer to AE ↓ or AEENDTC.

5. Expected Query Message:

- "Outcome suggests this Adverse event is still ongoing but End Date is entered. Please verify."

Field Information Breakdown:

- **Input Field 1:** AEOUT (Outcome)
 - Possible values: "Not recovered/not resolved", "Unknown", other values.
- **Input Field 2:** AEENDTC (End Date)
 - Value presence (i.e., empty vs. non-empty).
- **Expected Output:** Query message if the conditions are met.

Next Steps:

I will now generate the test data based on these conditions, including cases where values are empty.

[>]



Classic ECS and Test Plan Configuration



If **Outcome = Not recovered/not resolved OR Unknown AND End Date <> empty THEN fire query**



Test Plan Example

FormOID	Input Field 1 (Outcome)	Input Field 2 (End Date)	Expected FieldOID (Output)	Expected Output Value	Query Field	Query Message
AEG001	Not recovered/not resolved	2023/5/1	AEOUT	Query fired	AEOUT	Outcome suggests this Adverse event is still ongoing but End Date is entered. Please verify.
AEG001	Unknown	2023/6/15	AEOUT	Query fired	AEOUT	Outcome suggests this Adverse event is still ongoing but End Date is entered. Please verify.
AEG001	Recovered/resolved	2023/7/20	NA	No query	NA	NA
AEG001	Not recovered/not resolved		NA	No query	NA	NA



AI Service API Extension

Sample Code

Run assistant

```
1 # Create a thread
2 thread = client.beta.threads.create()
3
4 # Add a user question to the thread
5 message = client.beta.threads.messages.create(
6     thread_id=thread.id,
7     role="user",
8     content="Hi, Can you give me some examples of Edit check test plan?" # Replace this with your prompt
9 )
10
11
12
13 # Run the thread
14 run = client.beta.threads.runs.create(
15     thread_id=thread.id,
16     assistant_id=assistant.id
17 )
18
19 # Looping until the run completes or fails
20 while run.status in ['queued', 'in_progress', 'cancelling']:
21     time.sleep(1)
22     run = client.beta.threads.runs.retrieve(
23         thread_id=thread.id,
24         run_id=run.id
25     )
26
27 if run.status == 'completed':
28     messages = client.beta.threads.messages.list(
29         thread_id=thread.id
30     )
31     print(messages)
32 elif run.status == 'requires_action':
33     pretext = (Input) ...
```

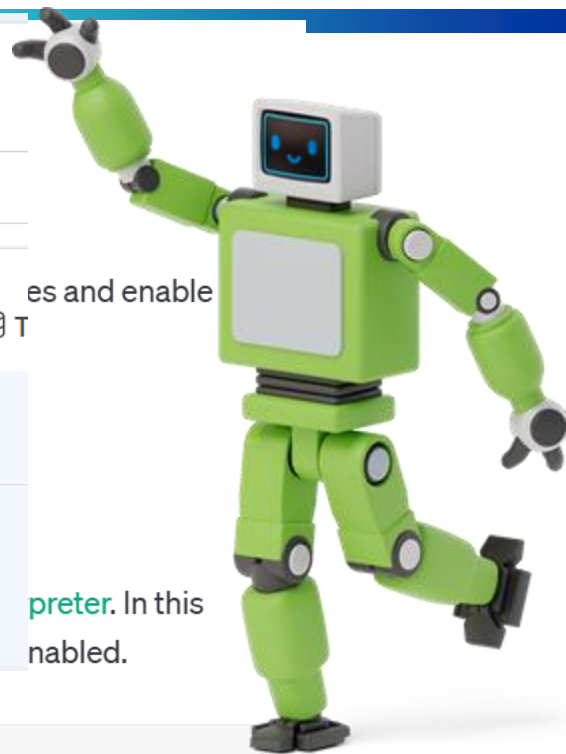
Search chat 🔍 Logs 📖 T

es and enable

prefer. In this
nabled.

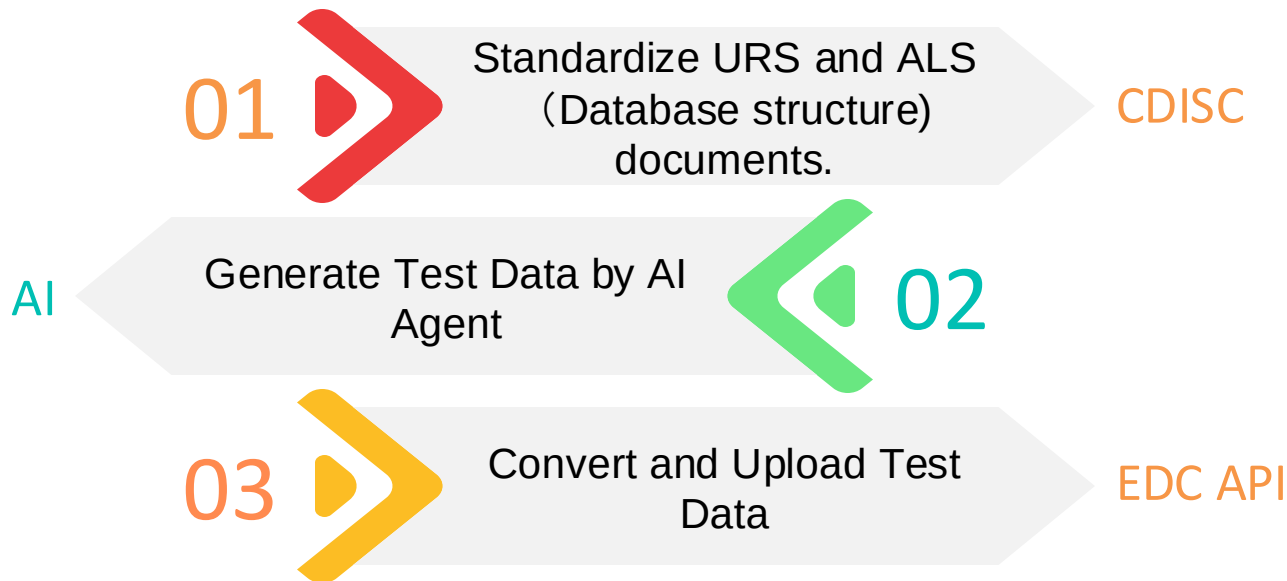
is handled automatically if you're using

cs=v2





AI-Driven URS Automation Process





Convert Test Data to EDC Model

```
data = u"""<?xml version="1.0" encoding="utf-8" ?>
<ODM CreationDateTime="2013-06-17T17:03:29" FileOID="3b9fea8b-e825-4e5f-bdc8-1464bdd7a664" FileType="ClinicalData"
  <ClinicalData MetaDataVersionOID="1.3" StudyOID="Mediflex(DEV)">
    <SubjectData SubjectKey="1001 005" TransactionType="Insert">
      <SiteRef LocationOID="NOV0001" />
      <StudyEventData StudyEventOID="SCREEN" TransactionType="Update">
        <FormData FormOID="DM">
          <ItemGroupData ItemGroupOID="DM">
            <ItemData ItemOID="AGE" Value="22" />
            <ItemData ItemOID="BRTHDTC" Value="1973 Jun 26" />
            <ItemData ItemOID="SEX" Value="FEMALE" />
          </ItemGroupData>
        </FormData>
      </StudyEventData>
    </SubjectData>
  </ClinicalData>
</ODM>"""
```



Upload the Test Data

Rave Developer Platform
Rave API
Rwslib

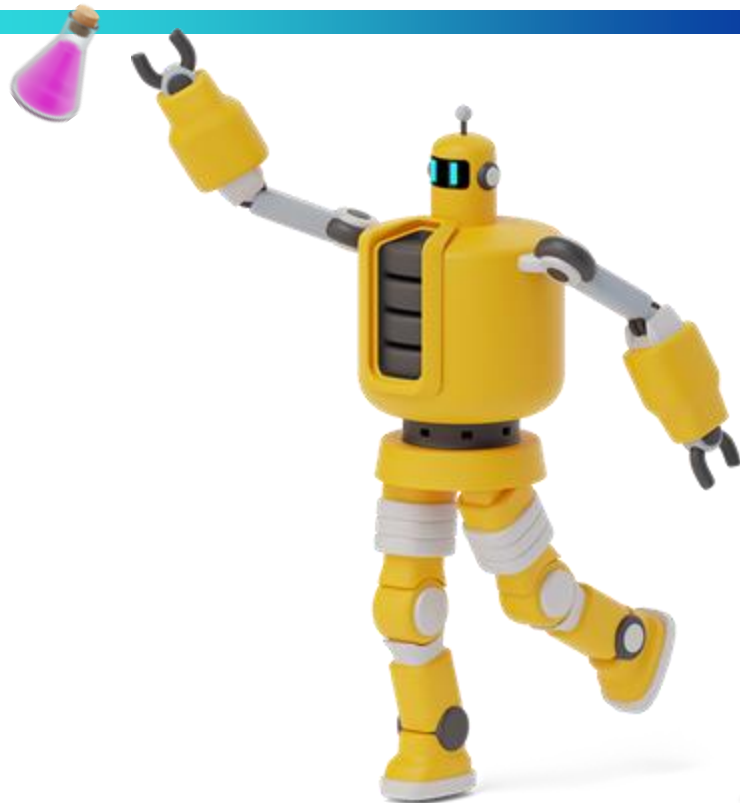
DataFormat

- ODM Format
- XML File

The screenshot shows a web interface for user selection. At the top, there is a search bar with the placeholder text "What are you looking for?". Below the search bar, there are three columns of filters:

I am a	I am looking for	About
☐ Auditor	☒ APIs	☐ Set up Study
☐ Clinical Research Associate	☒ API Resource	☐ Enroll Participants
☐ Clinical Research Coordinator	☒ Recipe	☐ Conduct Study
☐ Compliance Officer	☒ SDK	☐ Manage Study
☐ Data Manager	☐ Configuration	☐ Report and Manage Data
☐ Developer	☐ Implementation	☐ Close Study
☐ Monitor	☐ Instructions	
☐ Principal Investigator	☐ News	
☐ Site Administrator	☐ Overview	

At the bottom left of the filter panel, there is a link that says "Clear All Filters".





Rave API Functional Scope

- Rave Web Services (RWS) integrates Medidata Rave with third-party systems to exchange [CDISC ODM](#) standard clinical data or metadata synchronously and with immediate confirmation.
- <https://rwslib.readthedocs.io/en/latest/index.html>
- **Subject data (Clinical data) import**
- Subject data (Clinical data) export
- Study design (Metadata) import
- Study design (Metadata) export
- Custom dataset export
- Transaction logging
- Error reporting

How We Use Rave API

- [Study Metadata Endpoints](#)
- [Working with Clinical Data](#)
 - [Projectnames, Study names and Environments](#)
 - [ClinicalStudiesRequest\(\)](#)
 - [StudySubjectsRequest\(self, project_name, environment_name\)](#)
- [ODM Clinical View Datasets](#)
 - [StudyDatasetRequest\(project_name, environment_name\)](#)
 - [SubjectDatasetRequest\(project_name, environment_name, subjectkey\)](#)
 - [VersionDatasetRequest\(project_name, environment_name, version_oid\)](#)
 - [X-MWS-CV-Last-Updated](#)
- [Working with Architect Metadata](#)
 - [MetadataStudiesRequest\(\)](#)
 - [StudyDraftsRequest\(project_name\)](#)
 - [StudyVersionsRequest\(projectname\)](#)
 - [StudyVersionRequest\(projectname, oid\)](#)
 - [PostMetadataRequest\(projectname, data\)](#)
- [Working with Global Library Metadata](#)
 - [GlobalLibrariesRequest\(\)](#)
 - [GlobalLibraryDraftsRequest\(projectname\)](#)
 - [GlobalLibraryVersionsRequest\(projectname\)](#)
 - [GlobalLibraryVersionRequest\(projectname, version_oid\)](#)
- [Posting Clinical Data](#)
 - [PostDataRequest\(odm_data\)](#)
 - [Using Builders](#)
- [Using Builders](#)
 - [Metadata Builders](#)
- [Biostats Gateway Requests](#)
 - [CVMetaRequest\(project_name, environment_name\)](#)
 - [FormRequest\(project_name, environment_name, dataset_type, form_oid\)](#)
 - [MetaRequest\(\)](#)
 - [ProjectMetaRequest\(project_name\)](#)
 - [ViewMetaRequest\(view_name\)](#)
 - [CommentRequest\(project_name, environment_name\)](#)
 - [ProtocolDeviationsRequest\(project_name, environment_name\)](#)
 - [DataDictionariesRequest\(project_name, environment_name\)](#)
- [ODM Adapter Requests](#)
 - [AuditRecordsRequest\(project_name, environment_name\)](#)
 - [VersionFoldersRequest\(project_name, environment_name\)](#)
 - [SitesMetadataRequest\(\)](#)
 - [UsersRequest\(project_name, environment_name, location_oid=None\)](#)
 - [SignatureDefinitionsRequest\(project_name\)](#)

How We Use Clinflash API

```
EDC_API_KEY = os.getenv("EDC_API_KEY")

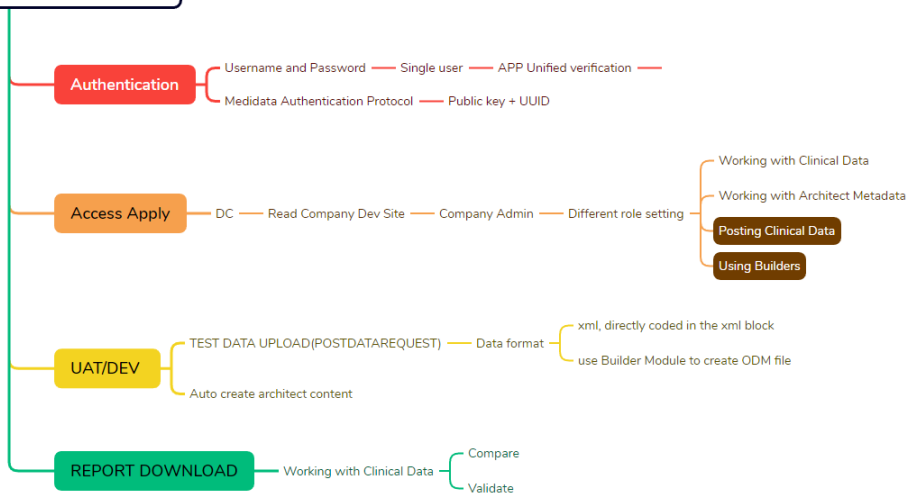
# 获取当前时间的毫秒数
def get_current_timestamp():
    return int(time.time() * 1000)

# 生成sign参数
def generate_sign(timestamp):
    raw_string = EDC_API_KEY + str(timestamp)
    return hashlib.md5(raw_string.encode()).hexdigest()

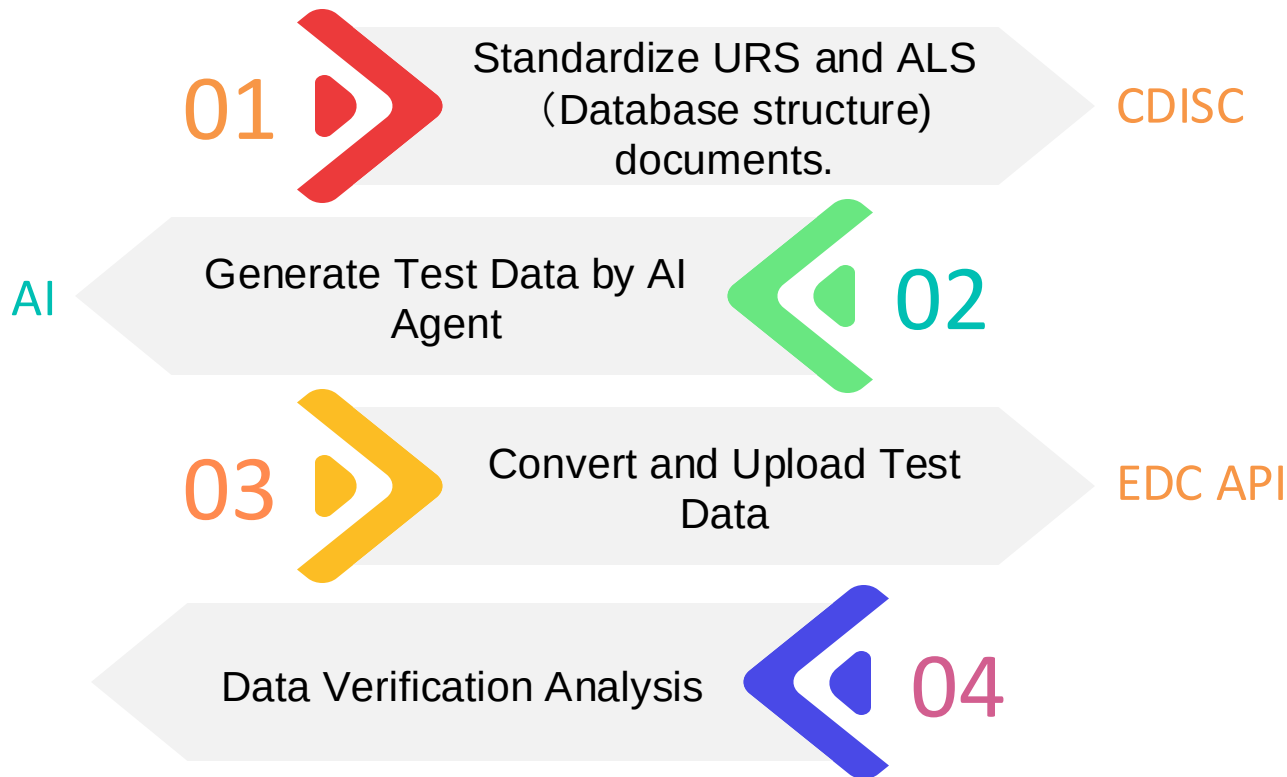
# 从API获取数据
def fetch_data_from_api(url, extra_params=None):
    timestamp = get_current_timestamp()
    sign = generate_sign(timestamp)
    data = {
        "timestamp": timestamp,
        "sign": sign
    }
    if extra_params:
        data.update(extra_params)
    response = requests.post(url, json=data)
    if response.status_code == 200 and response.json()["success"]:
        return response.json()["data"]
    else:
        print("Error fetching data from API:", response.json()["message"])
        return []

def convert_updateDT_in_dataframe(df):
    if 'updateDT' in df.columns:
        df['updateDT'] = pd.to_datetime(df['updateDT'], unit='ms')
    return df
```

Upload the UAT data via RAVE WEB API in DEV/UAT ENV



AI-Driven URS Automation Process



Data Verification Analysis



Step 01

Download the system reports

Step 02

Compare the results

Step 03

Identify the problematic EC

Step 04

Fix Programming Error



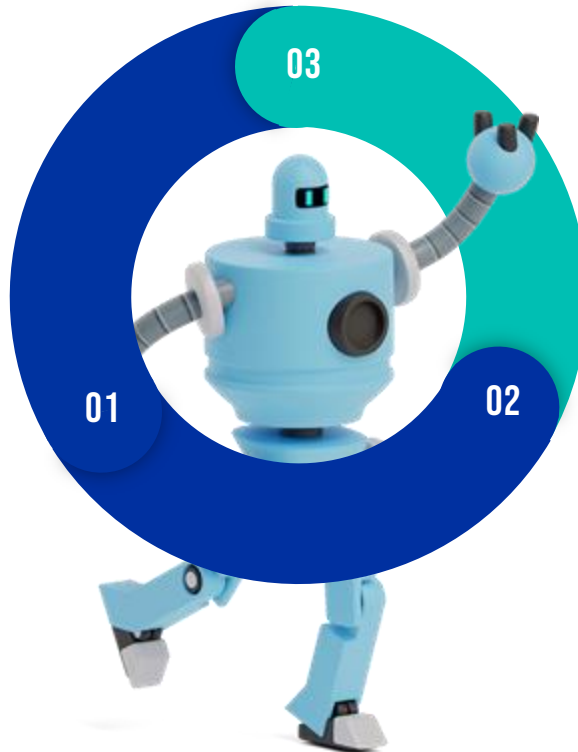
Data Verification Analysis

SubjectName	Folder	Form	Field	Log#	QryOpenBy	QueryText	MarkingGroupName	QueryStatus	Name
-------------	--------	------	-------	------	-----------	-----------	------------------	-------------	------



Standardized Data Throughout

**URS/ALS Identify and
Data Point Match**



**Query Report/Data
listing compare**

**ODM/XML Model
Established**



Strengths and Outlook

- Build Test Plan and Test Data Lib
 - Traceable
 - Reduce system interaction time
 - The larger the standardized metadata pool, the more accurate it is
- Improved Efficiency
 - Initial configuration consumes a lot of time, especially complex logic and cross-table logic, but once the initialization is completed, a lot of manual time will be saved later.



Strengths and Outlook

- Outlook
 - Automatically write Edit Checks
 - System self-check
 - GCP and regulatory requirements, how to verify the stability of systems and generated data



References

- **Rwslib**
 - Github Repo: <https://github.com/mdsol/rwslib>
 - Guide: <https://rwslib.readthedocs.io/en/latest/index.html>
- **Rave**
 - Developer Central: <https://www.medidata.com/en/clinical-trial-services/developer-central/>
 - Medidata Knowledge Hub : https://learn.medidata.com/en-US/auth/login?redirect=%2Fbundle%2Fmedidata%2Fpage%2Fmedidata_home.html

ANY QUESTIONS?



ACKNOWLEDGES

A sincere thank you to @Wei, Zhijun @ Tian, Chengeng @Yin, Xiaoran @Wang, Chengqian for their valuable assistance and thorough review.

Special thanks to **Guoyan.Zeng@Novartis.com** for the exceptional support and commitment throughout this process.

Thank you!

For further questions or feedback, please feel free to contact me:

Email: Flora-1.Chen@Novartis.com



**The Global Healthcare
Data Science Community**

Contact Channels

UK +44 1843 609600
office@phuse.global
phuse.global

Social Media

@phusetwitta
/phusebook
/phusetube
/company/phuse

THANK YOU!