

**MANASYS Jazz**

Leveraging the Past, Creating the Future

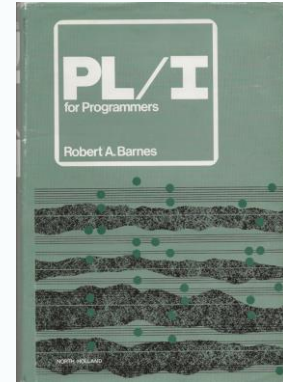
# Bridging Two Worlds

Presentation 10<sup>th</sup> May to  
Virtual CICS Users' Group



# Background

- Who am I?
  - 1970 – Mainframe Developer
    - Databank Systems Ltd
    - Air New Zealand
    - PL/I for Programmers (1979)
    - Encountered Easytrieve



# Background

- Who am I?
  - 1970 – Mainframe Developer
  - 1980 – MANA Systems Ltd
    - MANASYS V1
      - Easytrieve generating PL/I or COBOL
    - MANASYS Fujitsu
      - Fujitsu Strategic Software Product
      - Relational Databases
      - ~3270-type on-line systems
      - No PL/I Option

# Background

- Who am I?
  - 1970 – Mainframe Developer
  - 1980 – MANA Systems Ltd
    - MANASYS V1 (Easytrieve generating PL/I or COBOL)
    - MANASYS Fujitsu (Strategic Software Product)
  - 1996 – IT Consultant
    - “What every unemployed IT Professional calls themselves”
    - Several Successful IT Projects

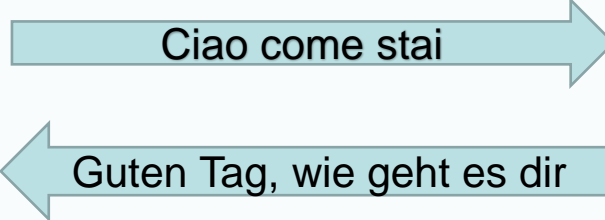
# Background

- Who am I?
  - 1970 – Mainframe Developer
  - 1980 – MANA Systems Ltd
    - MANASYS V1 (Easytrieve generating PL/I or COBOL)
    - MANASYS Fujitsu (Strategic Software Product)
  - 1996 – IT Consultant
  - 2013 – Jazz Software Ltd
    - MANASYS Jazz

## Bridging these different Worlds needs CICS Web Services and client apps



Client  
Apps



# CICS Web Services

- Like a Classical CICS program
  - Input msg, do something, output msg
- But different
  - Messages not 3270 screens – any length, any format.
  - No COMMAREA, so everything must be in the message
  - Communicate with HTTP in a standard format (JSON) that can be understood by client programs whatever their technology
  - Clients aren't just displays – have their own logic

# How they work

REST, not SOAP

JSON, not WSDL

What we write  
(COBOL, PL/I etc)

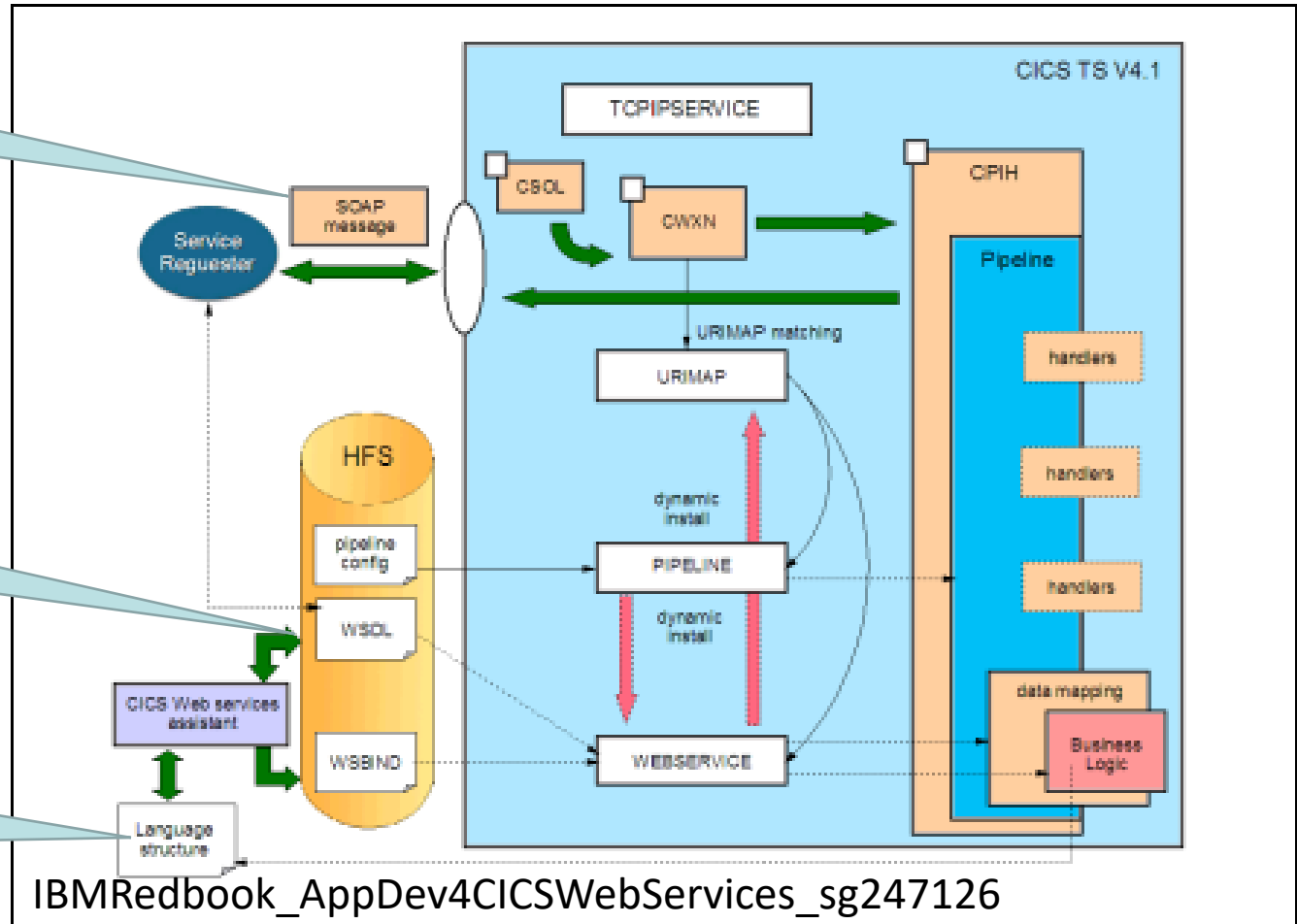


Figure 2-1 Web service runtime service provider processing



# JSPG2

## Update EMPLOYEE Records

```
db2 => connect to sample

Database Connection Information

Database server      = DB2/NT64 11.1.3.3
SQL authorization ID = ROBERTBW...
Local database alias = SAMPLE

db2 => select * from Employee
```

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM	CURRENCY	DEPTMGR
000010	CHRISTINE	I	HAAS	A00	3978	01/01/1995	PRES	18	F	08/24/1963	160387.00	1000.00	4220.00	USD	N
000020	MICHAEL	L	THOMPSON	B01	3476	10/10/2003	MANAGER	18	M	02/02/1978	94250.00	800.00	3300.00	USD	Y
000030	SALLY	A	KWAN	C01	4738	04/05/2005	MANAGER	20	F	05/11/1971	98250.00	800.00	3060.00	USD	Y
000050	JOHN	B	GEYER	E01	6789	08/17/1979	MANAGER	16	M	09/15/1955	80175.00	800.00	3214.00	USD	Y
100321															
000070															
000090															
000100															
000110															
000120															
000130															
000140															
000150	Robert		ADAMSON	D21	2468	02/12/2002									
000160	Mary	E	Pym	D11	2618	10/11/2006									
000045	IRVING	N	STERN	D11	2618	09/14/2005									
000180	MARILYN	S	SCOUTTEN	D11	1682	07/07/2003									
000190	JAMES	H	WALKER	D11	2986	07/26/2004									
000200	DAVID		BROWN	D11	4501	03/03/2002									
000210	WILLIAM	T	JONES	D11	0942	04/11/1998	DESIGNER	17	M	02/23/2003	68270.00	400.00	1462.00	USD	N
000220	JENNIFER	K	LUTZ	D11	0672	08/29/1998	DESIGNER	18	F	03/19/1978	49840.00	600.00	2387.00	USD	N
000230	JAMES														
000240	SALVATORE														
000250	DANIEL														
000260	SYBIL														
000270	MARIA														
000280	ETHEL														
000290	JOHN														
000300	PHILIP														
000310	MAUDE	F	SETRIGHT	E11	3332	09/12/1994	OPERATOR	12	F	04/21/1961	35900.00	300.00	1272.00	USD	N

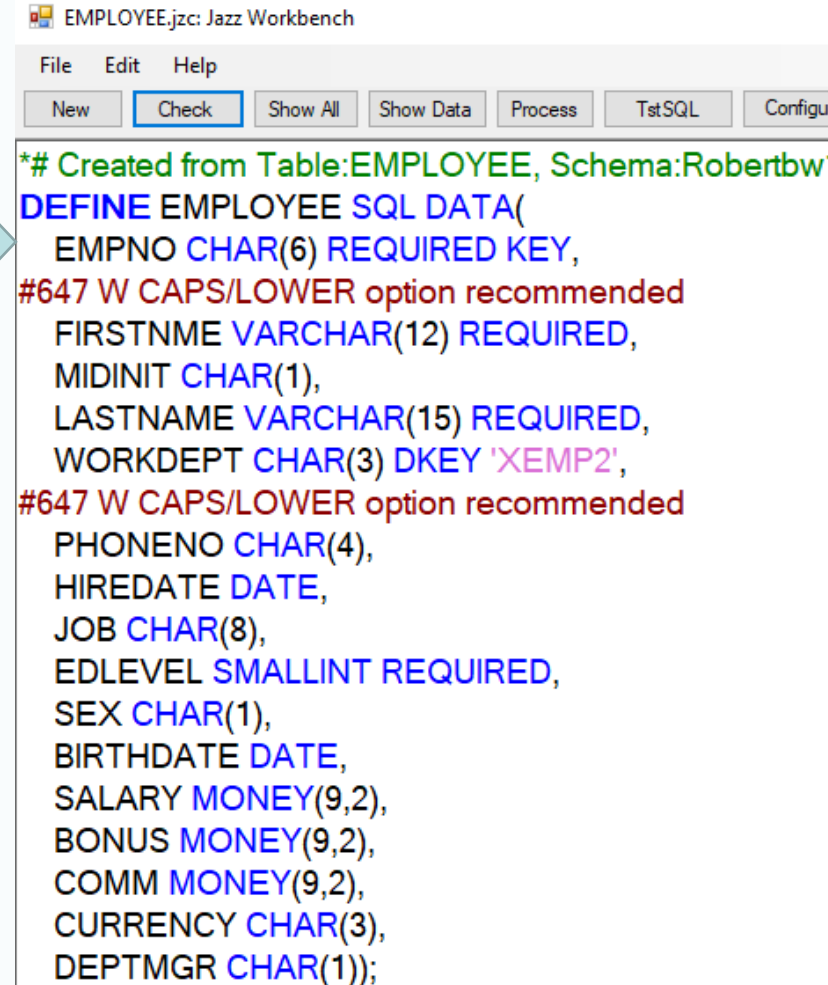
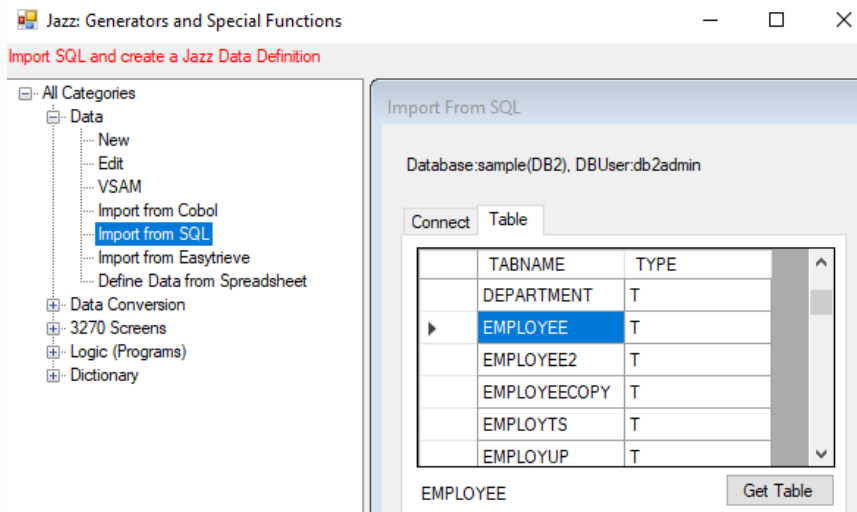
1. Need to "Lock" record for update

2. Check input – Numeric, Sex M or F, Dates, etc

3. Plus check I/O, formatting, etc

Even for something this simple, Expect 000's of COBOL loc

# Define EMPLOYEE from database



But there's a lot of missing  
information

# Program JSPG2

## Update EMPLOYEE Record

\*# Last Updated by JAZZUSR at 19/06/2021 10:38:02 AM

\*# Created from Table:EMPLOYEE, Schema:ROBERTBW10, Database:Sample by JazzUser at 26/03/2019 4:2 PM

**COPY** Department;

```
DEFINE EMPLOYEE SQL DATA(  
  EMPNO CHAR(6) PIC '999999' REQUIRED KEY,  
  FIRSTNME VARCHAR(12) REQUIRED,  
  MIDINIT CHAR (1),  
  LASTNAME VARCHAR(15) REQUIRED,  
  WORKDEPT CHAR(3) CAPS DKEY 'XEMP2' EXISTS DEPARTMENT.DEPTNO,  
  PHONENO CHAR(4) PIC '9999',  
  HIREDATE DATE,  
  JOB CHAR(8),  
  EDLEVEL SMALLINT REQUIRED,  
  SEX CHAR(1) CAPS CODES(M:Male, F:Female),  
  BIRTHDATE DATE,  
  SALARY MONEY(9,2) MIN 0,  
  BONUS MONEY(9,2) MIN 0,  
  COMM MONEY(9,2) MIN 0,  
  CURRENCY CHAR(3),  
  DEPTMGR BOOLEAN);
```

Edited to strengthen  
validation

# Generate JSPG2 (Demo, Generate Dialog)

**1**  
Web Service

**2**  
Give Details

**3**  
Finish

Jazz: Generators and Special Functions

All Categories

- Data
- Data Conversion
- 3270 Screens
- Logic (Programs)
  - Batch Program
  - CICS Program
  - Report Designer
  - Web Service**
  - SubProgram
  - Edit
- Dictionary

**New Web Service**

Type  
 WSDL  JSON

Program Name: JSPG1  Replace

Service: MyJSv

Container: DFHWS-DATA  Replace  Reuse

Use Database sample

Table 1: Employee  Update Max: 10  Add  Del

Keys  All Fields  Select

Table 2:

URL  
@Swagger

Host: localhost

Port: 9003

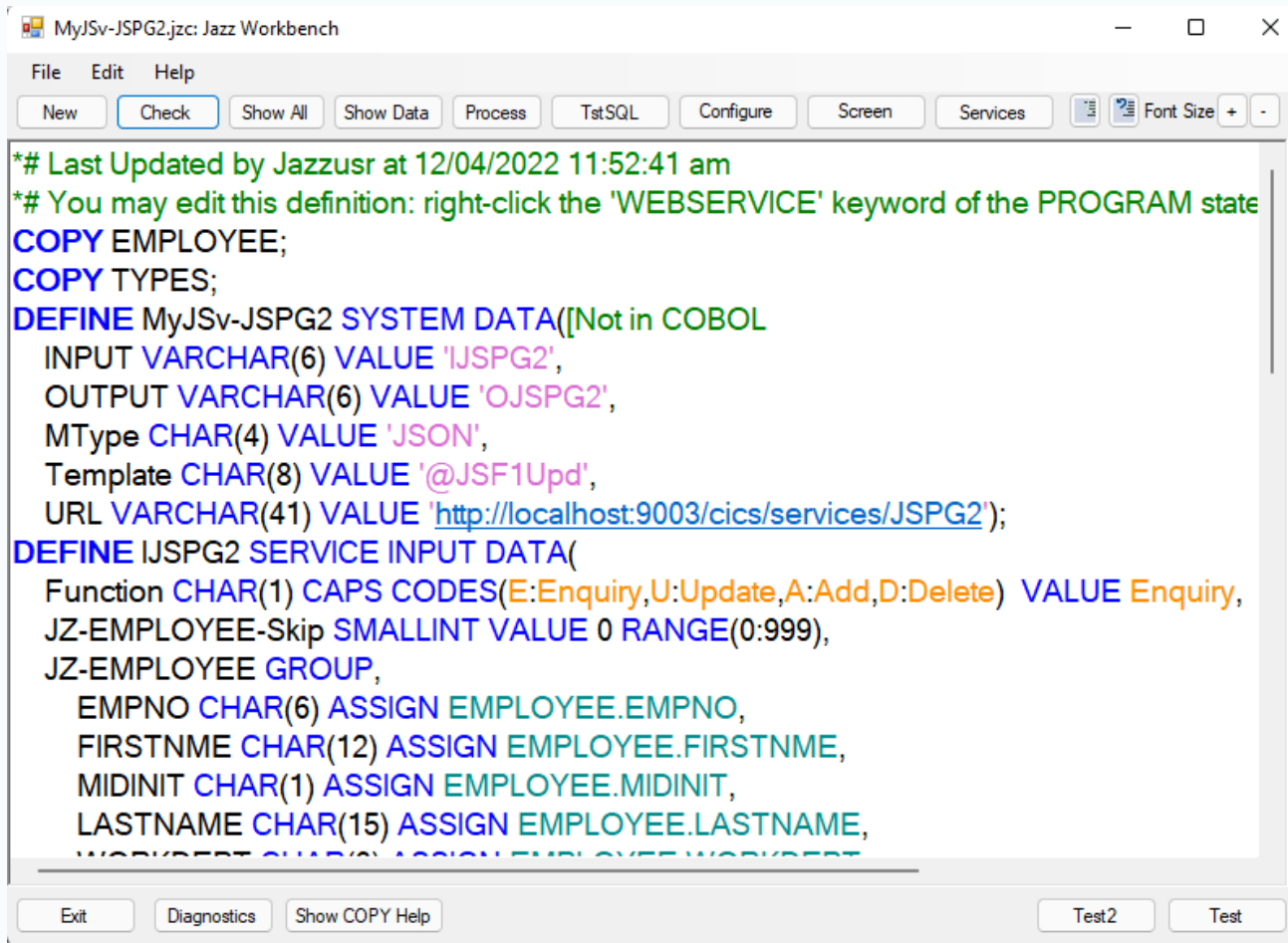
basePath: /cics/services

Scheme  
 HTTP  HTTPS

Template: @JSF1Upd

Finish Cancel

# Generate JSPG2 (Demo, Message Defn)



The screenshot shows the Jazz Workbench interface for a program named 'MyJSv-JSPG2.jzc'. The main window displays the following code:

```
*# Last Updated by Jazzusr at 12/04/2022 11:52:41 am
*# You may edit this definition: right-click the 'WEBSERVICE' keyword of the PROGRAM state
COPY EMPLOYEE;
COPY TYPES;
DEFINE MyJSv-JSPG2 SYSTEM DATA([Not in COBOL
  INPUT VARCHAR(6) VALUE 'JSPG2',
  OUTPUT VARCHAR(6) VALUE 'OJSPG2',
  MType CHAR(4) VALUE 'JSON',
  Template CHAR(8) VALUE '@JSF1Upd',
  URL VARCHAR(41) VALUE 'http://localhost:9003/cics/services/JSPG2');
DEFINE IJSPG2 SERVICE INPUT DATA(
  Function CHAR(1) CAPS CODES(E:Enquiry,U:Update,A:Add,D>Delete) VALUE Enquiry,
  JZ-EMPLOYEE-Skip SMALLINT VALUE 0 RANGE(0:999),
  JZ-EMPLOYEE GROUP,
  EMPNO CHAR(6) ASSIGN EMPLOYEE.EMPNO,
  FIRSTNME CHAR(12) ASSIGN EMPLOYEE.FIRSTNME,
  MIDINIT CHAR(1) ASSIGN EMPLOYEE.MIDINIT,
  LASTNAME CHAR(15) ASSIGN EMPLOYEE.LASTNAME,
```

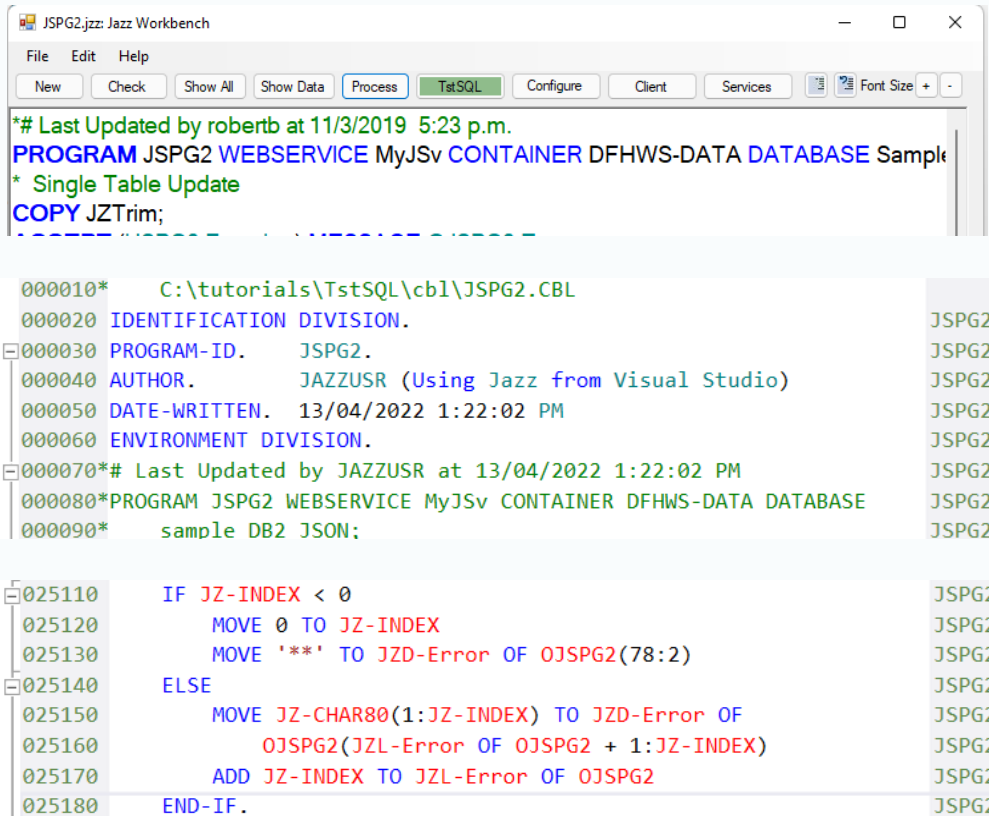
# Generate JSPG2 (Demo, Logic)

```
JSPG2.jzz: Jazz Workbench
File Edit Help
New Check Show All Show Data Process TestSQL Configure Client Services Font Size + -
*# Last Updated by robertb at 11/3/2019 5:23 p.m.
PROGRAM JSPG2 WEBSERVICE MyJSv CONTAINER DFHWS-DATA DATABASE Sample DB2
* Single Table Update
COPY JZTrim;
ACCEPT (JSPG2.Function) MESSAGE OJSPG2.Error;
#052 W Item IJSPG2.Function will be validated, but not moved from the input record
CASE (IJSPG2.Function);
  WHEN (Enquiry);
    ACCEPT (EMPLOYEE.EMPNO = IJSPG2.EMPNO OR EMPLOYEE.WORKDEPT = IJSPG2.WORKDEPT) MESSAGE OJSPG2.Error;
    GET EMPLOYEE KEY(EMPLOYEE.EMPNO OR EMPLOYEE.WORKDEPT) SAVESUM OJSPG2.JZ-EMPLOYEE.Checksum;
    #628 W GENERIC assumed for WORKDEPT
    END GET EMPLOYEE RESPOND OJSPG2;
  WHEN (Update);
    ACCEPT (EMPLOYEE.EMPNO=IJSPG2.EMPNO) MESSAGE OJSPG2.Error;
    GET EMPLOYEE KEY(EMPLOYEE.EMPNO) UPDATE CHECKSUM IJSPG2.JZ-EMPLOYEE.Checksum;
    COPY JZSMth;
    COPY JZMDays;
    COPY JZDTS;
    ACCEPT (IJSPG2.JZ-EMPLOYEE.*) EXCEPT(EMPLOYEE.EMPNO) MESSAGE OJSPG2.Error;
    #207 I JZ-EMPLOYEE.FIRSTNAME,JZ-EMPLOYEE.MIDINIT,JZ-EMPLOYEE.LASTNAME,JZ-EMPLOYEE.WORKDEPT,JZ-EMPLOYEE
    END GET EMPLOYEE UPDATE RESPOND OJSPG2;
  WHEN (Add);
    ACCEPT (EMPLOYEE.EMPNO = IJSPG2.EMPNO) MESSAGE OJSPG2.Error;
```

ACCEPT  
Validates  
input for  
definition,  
Assigns it

SAVESUM  
CHECKSUM  
Manage CICS  
Pseudo-Locking

# Generate JSPG2 (Demo, COBOL)



```
File Edit Help
New Check Show All Show Data Process TestSQL Configure Client Services Font Size + -

*# Last Updated by robertb at 11/3/2019 5:23 p.m.
PROGRAM JSPG2 WEBSERVICE MyJSv CONTAINER DFHWS-DATA DATABASE Sample
* Single Table Update
COPY JZTrim;

000010* C:\tutorials\TstSQL\cbl\JSPG2.CBL
000020 IDENTIFICATION DIVISION. JSPG2
000030 PROGRAM-ID. JSPG2. JSPG2
000040 AUTHOR. JAZZUSR (Using Jazz from Visual Studio) JSPG2
000050 DATE-WRITTEN. 13/04/2022 1:22:02 PM JSPG2
000060 ENVIRONMENT DIVISION. JSPG2
000070*# Last Updated by JAZZUSR at 13/04/2022 1:22:02 PM JSPG2
000080*PROGRAM JSPG2 WEBSERVICE MyJSv CONTAINER DFHWS-DATA DATABASE JSPG2
000090* sample DB2 JSON; JSPG2

025110 IF JZ-INDEX < 0 JSPG2
025120 MOVE 0 TO JZ-INDEX JSPG2
025130 MOVE '***' TO JZD-Error OF OJSPG2(78:2) JSPG2
025140 ELSE JSPG2
025150 MOVE JZ-CHAR80(1:JZ-INDEX) TO JZD-Error OF JSPG2
025160 OJSPG2(JZL-Error OF OJSPG2 + 1:JZ-INDEX) JSPG2
025170 ADD JZ-INDEX TO JZL-Error OF OJSPG2 JSPG2
025180 END-IF. JSPG2
```

Click [Process].

In less than 1 second,  
2500+ lines of COBOL  
generated

Z/OS – JCL compiles the  
program ✓ and generates  
the JSON. Install in  
CICS, and it runs  
correctly first time ✓

Micro Focus – similar, but  
uses MFED dialogs

# Test (ReadyAPI)

JSON Node	Value
[-] JSPG2	
[-] IJSPG2	
JZ_Function	E
JZ_Employee_Skip	0
[-] JZ_Employee	
EMPNO	60
FIRSTNME	
MIDINIT	
LASTNAME	
WORKDEPT	
PHONENO	
HIREDATE	
JOB	
EDLEVEL	
SEX	
BIRTHDATE	
SALARY	
BONUS	
COMM	
JZ_CURRENCY	
[-] ViewState	
Checksum_Employee	

JSON Node	Value
[-] JSPG2Response	
[-] OJSPG2	
[-] JZ_ERROR	
JZL_ERROR	0
JZD_ERROR	
[-] ViewState	
Checksum_Employee	6E5A155E6766F04B7BBC30EE6AC7FD8FC719BC3F
JZ_Employee_ReadTo	0
JZ_Employee_NbrReturned	1
JZ_Employee_ReturnCode	
[-] JZ_Employee	
EMPNO	000060
[-] FIRSTNME	
JZL_FIRSTNME	6
JZD_FIRSTNME	IRVING
MIDINIT	F
[-] LASTNAME	
JZL_LASTNAME	5
JZD_LASTNAME	STERN
WORKDEPT	D11
PHONENO	6423
HIREDATE	20030914
JOB	MANAGER
EDLEVEL	16
SEX	M
BIRTHDATE	19750707
SALARY	72250
BONUS	500
COMM	2580
JZ_CURRENCY	USD



# Bridging these different Worlds needs CICS Web Services and **client apps**



Client

Apps



Must work together  
Using different tools  
C#, Java | COBOL

Ciao come stai

Guten Tag, wie geht es dir



## Interface JSPG2Client



Client  
Apps



### Interface JSPG2Client

C#, .NET CORE  
=> Windows, Linux,  
Android, IOS, ...

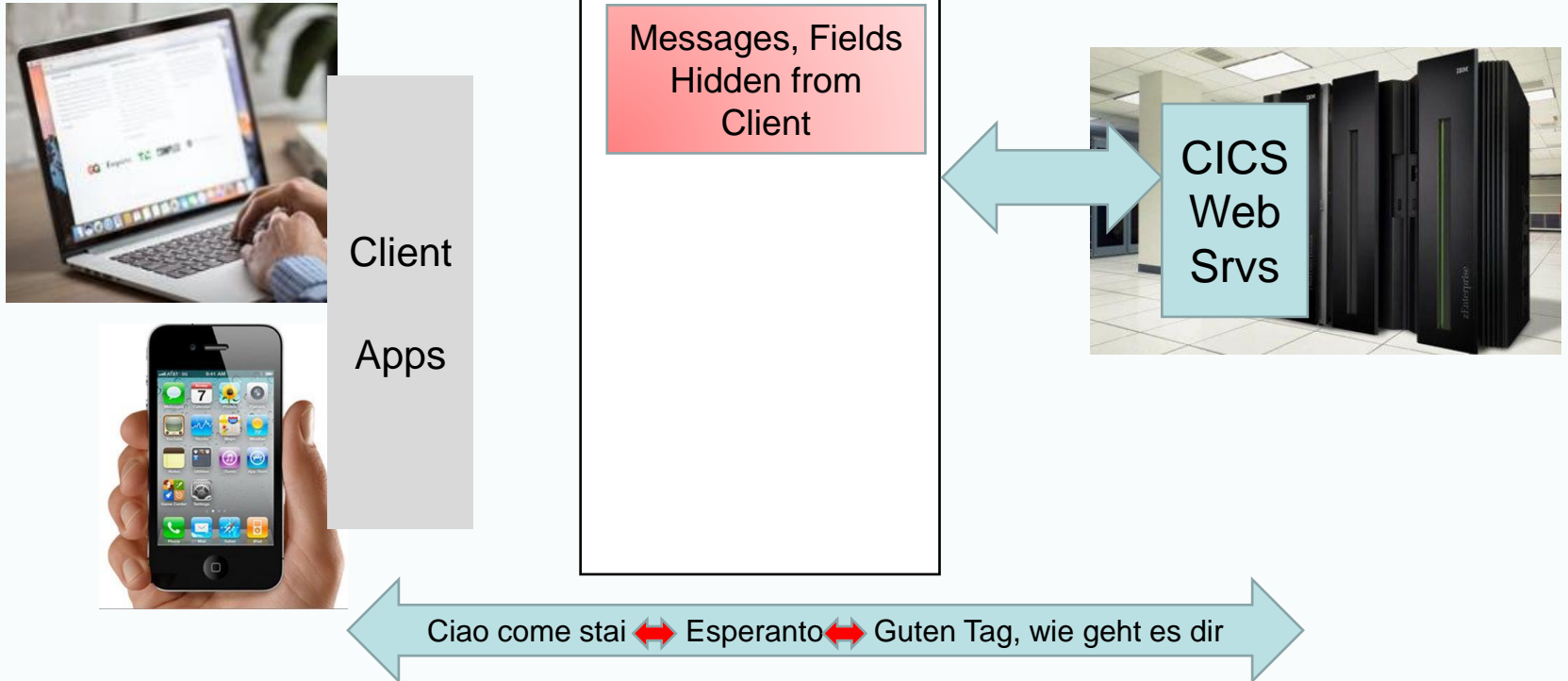
#### Service Rules

- Field validity
- Method arguments
- Sequencing

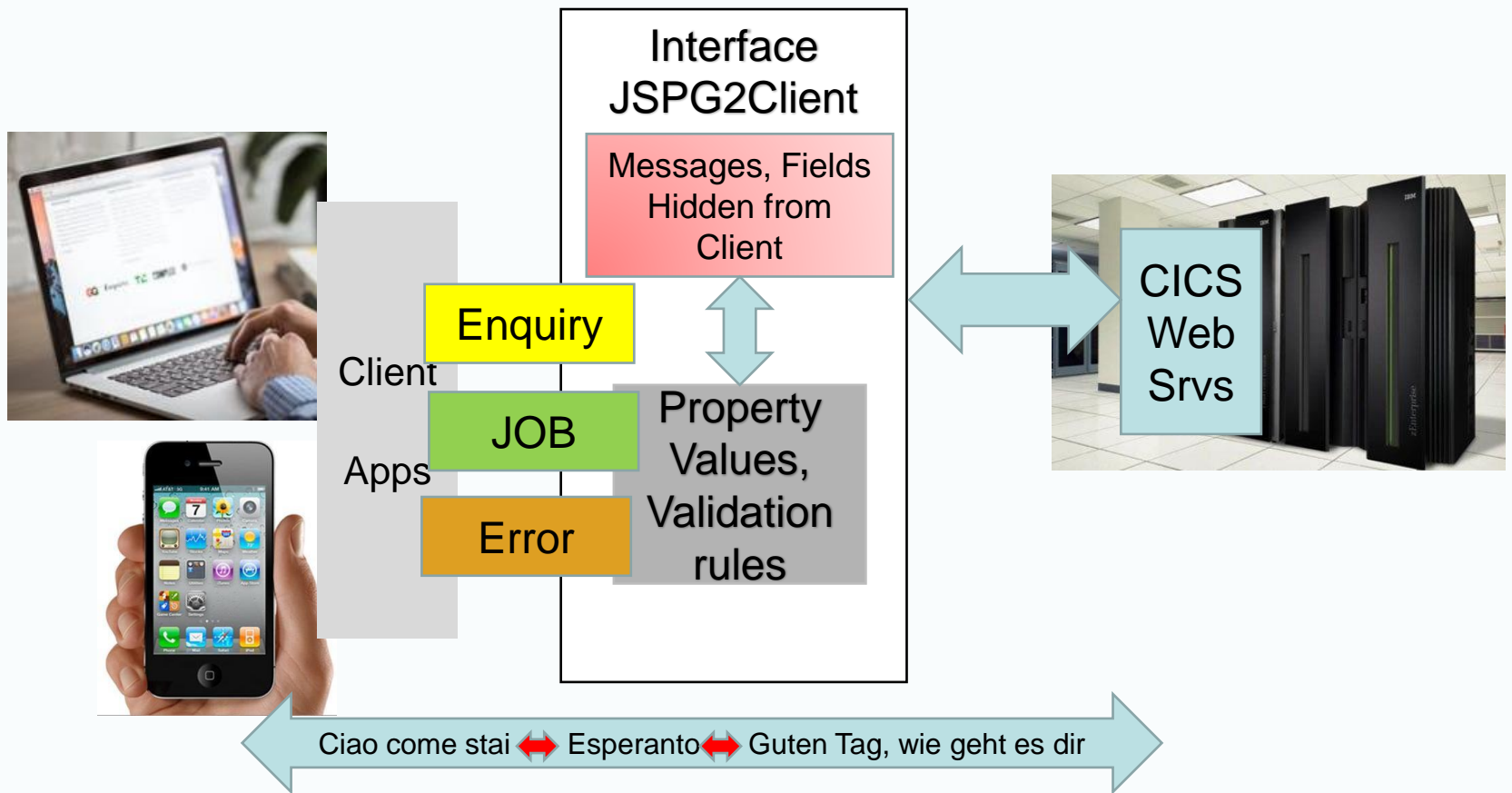


Ciao come stai ↔ Esperanto ↔ Guten Tag, wie geht es dir

## Interface JSPG2Client



## Interface JSPG2Client



# Create an Interface

<https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm>

## 1. Once only: configure Jazz to create interfaces

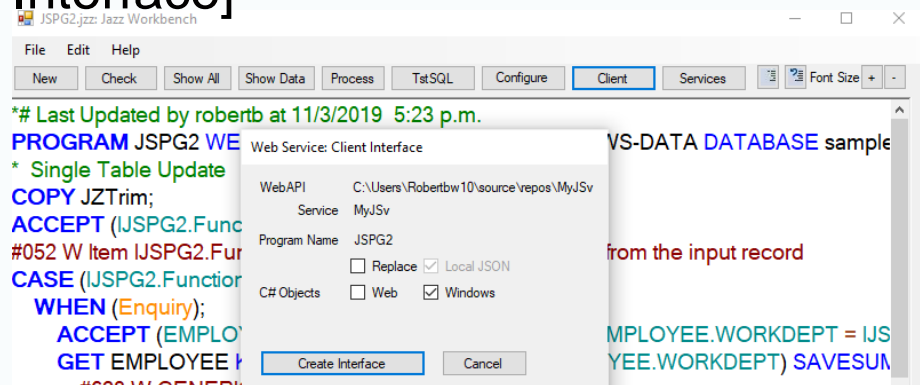
Create an Interface Project (MyJSv), Configure Jazz to use this project

## 2. Create Interface. With the Jazz Workbench

1. Open the program (e.g. JSPG2). Check that its been compiled and bound, and that the JSON in project's root folder (NB zOS!)

2. Click [Client], then [Create Interface]

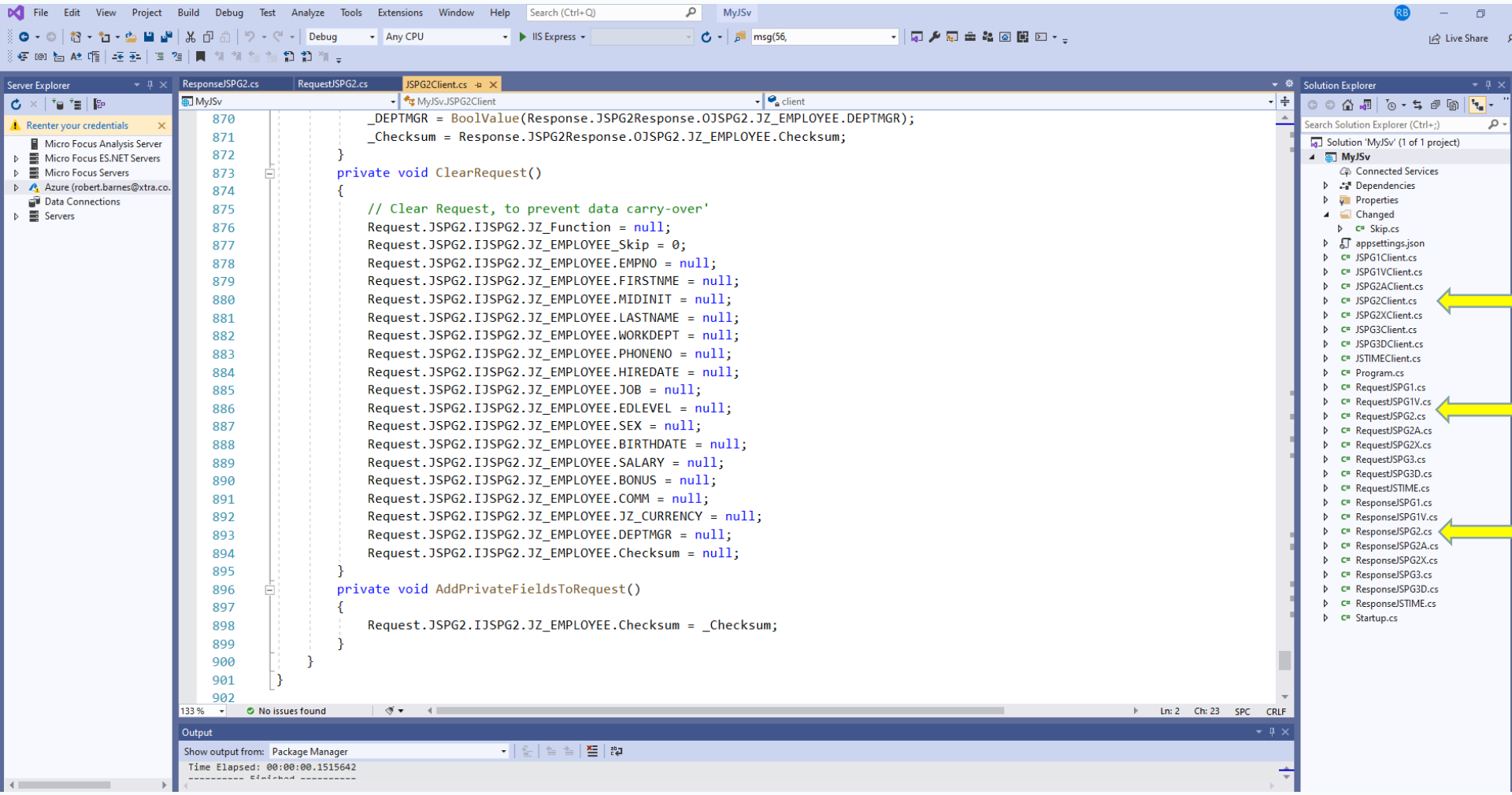
=> C# Interface is created  
(~1000 lines for JSPG2)



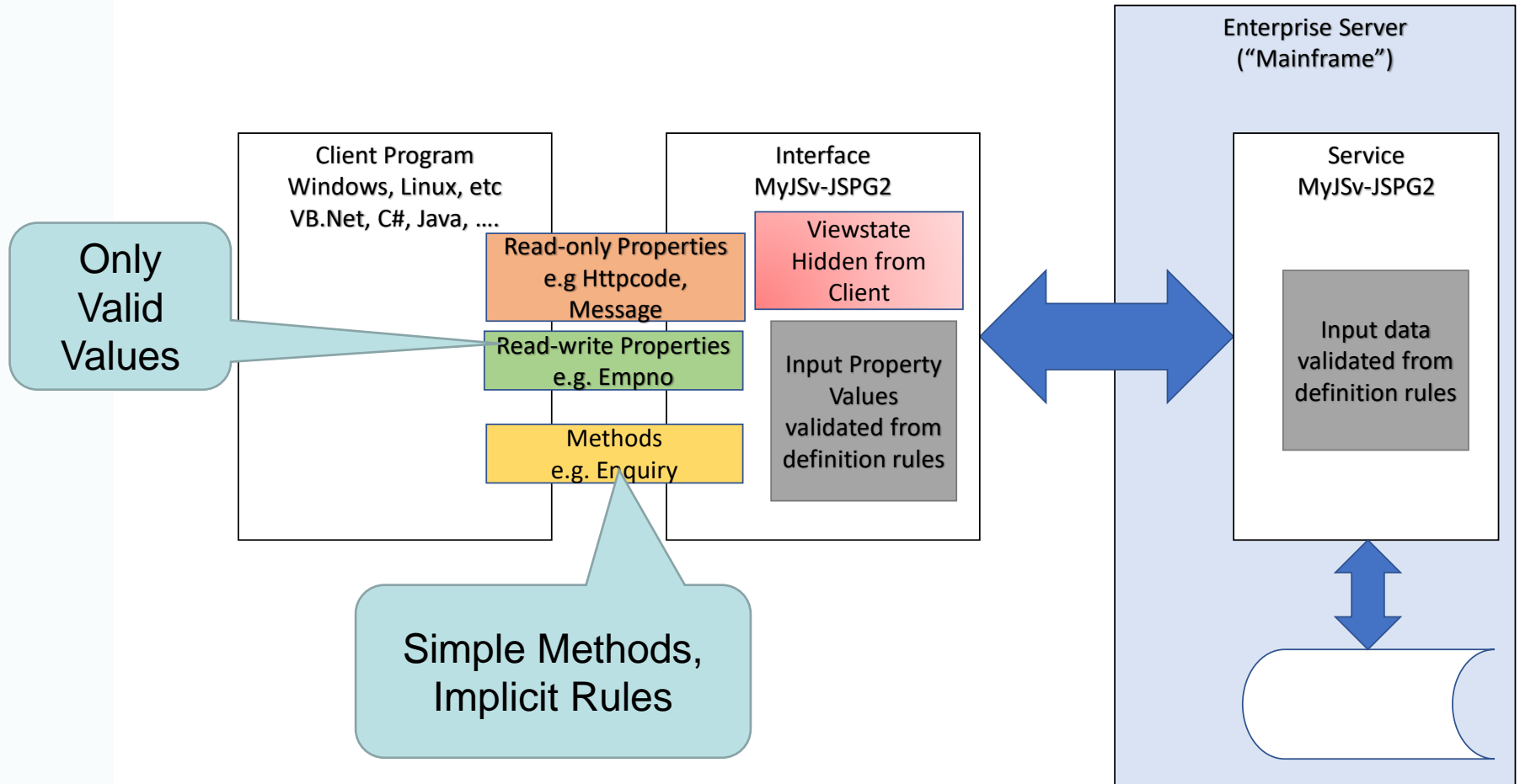
For example, see

<https://www.jazzsoftware.co.nz/Docs/JSPG2AClient.cs.txt>

# JSPG2Client (Interface)



# Create a Client



[https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#\\_Toc101020002](https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#_Toc101020002)

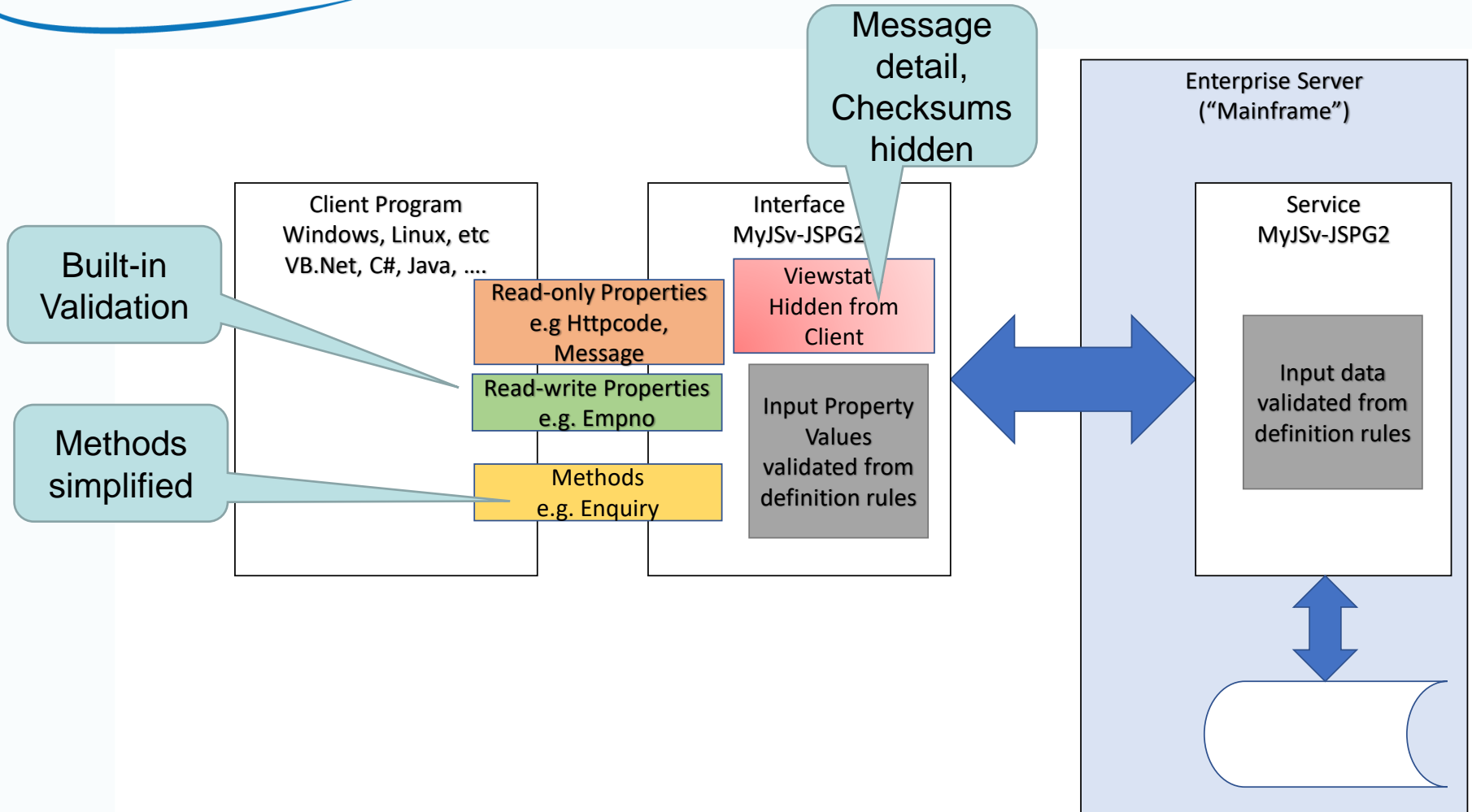
# JSPG2Test

The screenshot shows the JSPG2Test application window with a red "IblError" message at the top. The form contains several input fields and buttons. The EMPNO field is highlighted in cyan, and the WORKDEPT field is highlighted in green. The form includes a scrolling control with buttons for navigation and a "Skip" button. Below the form are buttons for "Enquiry", "Update", "Add", "Delete", "Clear", and "Close". The form fields are: FIRSTNAME, MIDINIT, LASTNAME, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM, CURRENCY, and DEPTMGR. The HTTP Response and IblRawResponse fields are empty.

The screenshot shows the JSPG2Test application window with a red "SQL UPDATE/INSERT SUCCESSFUL" message at the top. The form contains several input fields and buttons. The EMPNO field is highlighted in cyan with the value "000010", and the WORKDEPT field is highlighted in green with the value "A00". The form includes a scrolling control with buttons for navigation and a "Skip" button. Below the form are buttons for "Enquiry", "Update", "Add", "Delete", "Clear", and "Close". The form fields are: FIRSTNAME (CHRISTOPHER), MIDINIT (I), LASTNAME (HAAS), PHONENO (3978), HIREDATE (1/01/1995), JOB (PRES), EDLEVEL (18), SEX (M), BIRTHDATE (24/08/1963), SALARY (\$160,387.00), BONUS (\$1,000.00), COMM (\$4,220.00), CURRENCY (USD), and DEPTMGR (True). The HTTP Response is "200(OK), Success:True, Reason:OK".

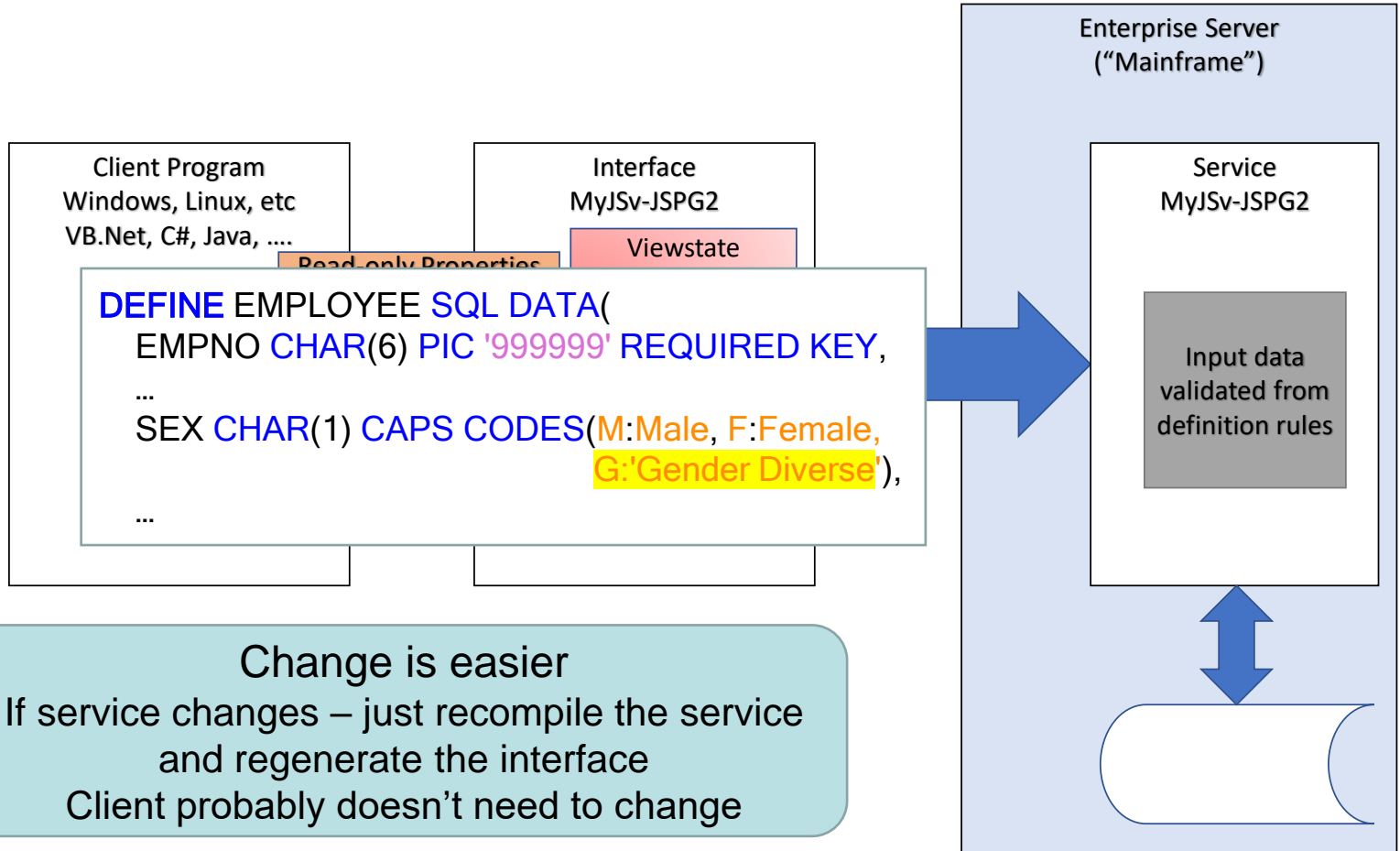


# Review



[https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#\\_Toc101020002](https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#_Toc101020002)

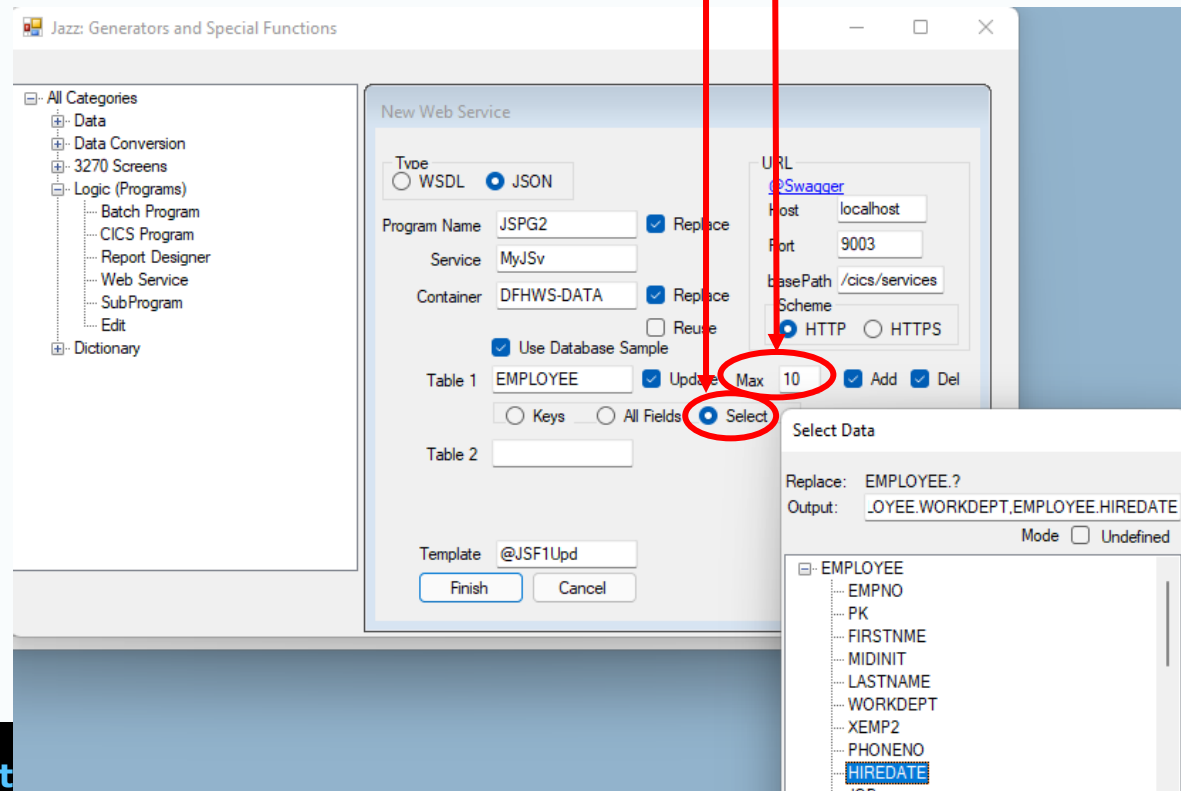
# Review



[https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#\\_Toc101020002](https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#_Toc101020002)

# Already Available

- Generate CICS Web Services for
  - VSAM or DB2
    - One or Many Records per Request/Response
    - Messages with Full Record, or Selected Fields



# Already Available

- Generate CICS Web Services for
  - VSAM or DB2
    - Full Record, or Selected Fields
    - One or Many Records per Request/Response
  - Single record type, or Parent Child
    - Plus related records
    - Enquiry, Update, Add Delete functions

# Already Available

- **Generate CICS Web Services for**

- VSAM or DB2

- Full Record, or Selected Fields
- One or Many Records per Request/Response

- Single record type, or Parent Child

- Plus related records
- Enquiry, Update, Add Delete functions

- **Multi-level Logic**

- Generate/Jazz Logic/COBOL

- Calculations of any complexity, CALL subprograms, etc

# Already Available

- **Generate CICS Web Services for**
  - VSAM or DB2
    - Full Record, or Selected Fields
    - One or Many Records per Request/Response
  - Single record type, or Parent Child
    - Plus related records
    - Enquiry, Update, Add Delete functions
- **Multi-level Logic**
  - Generate/Jazz Logic/COBOL
- **Two-click Interface Generation**

# Potential Futures

- Customers will set our priorities
  - They have a better idea of what's important
- Some ideas in Road Map

[https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#\\_Toc101020047](https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm#_Toc101020047)

- Conversations (Viewstate)
- More flexible data structures
- Prototype clients

# More Information

<https://jazzsoftware.co.nz/>

General information, download software, help and training

See <https://www.jazzsoftware.co.nz/Videos/bridge/bridge.html>

– Other Videos

- Batch – Reporting, EZT conversion, Data mapping and conversion
- Classical CICS development
- SOAP (WSDL) Web services

– Help and Training <https://jazzsoftware.co.nz/Help.aspx>

- Video Tutorials
- Help
  - Language Reference
  - Users' Guide
  - Publications
  - etc



# More Information

<https://jazzsoftware.co.nz/>

General information, download software, help and training

See <https://www.jazzsoftware.co.nz/Videos/bridge/bridge.html>

- **Web Services**

- **Creating Web Services**

- <https://www.jazzsoftware.co.nz/Docs/JazzUGMFStep4.htm>

- **Client Interfaces**

- <https://www.jazzsoftware.co.nz/Docs/JazzUGClientInt.htm>

# Thank you for listening to me

Email [Robert@Jazzsoftware.co.nz](mailto:Robert@Jazzsoftware.co.nz)



LinkedIn [linkedin.com/in/robert-barnes-5b833a](https://www.linkedin.com/in/robert-barnes-5b833a)

## Any Questions?