



IBM Software Group

*27th ALCS User Group Meeting, London, 24 - 26 May 2011*

# ALCS WAS - OLA

Nico Wigmans  
*alcs@uk.ibm.com*



# Legal Notices

**IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.**

**THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:**

- **CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS);**
- OR**
- **ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE**

## Trademarks

**The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.**

RACF*	DB2*	Hiperspace
IBM*	IBM logo*	MVS
Websphere*	VTAM*	z/Architecture
z/OS*	zSeries*	

\* Registered trademarks of IBM Corporation

**The following are trademarks or registered trademarks of other companies.**

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

**Other company, product and service names may be trademarks or service marks of others.**

# WebSphere Application Server (WAS)

- WebSphere Application Server z/OS
- Robust and modern “glue” system
- Can be a “front-end” to CICS, IMS, ALCS
- zAAP processors
  - Lower cost of acquisition
  - Do not contribute to software MLC
  - Competitive with distributed platforms



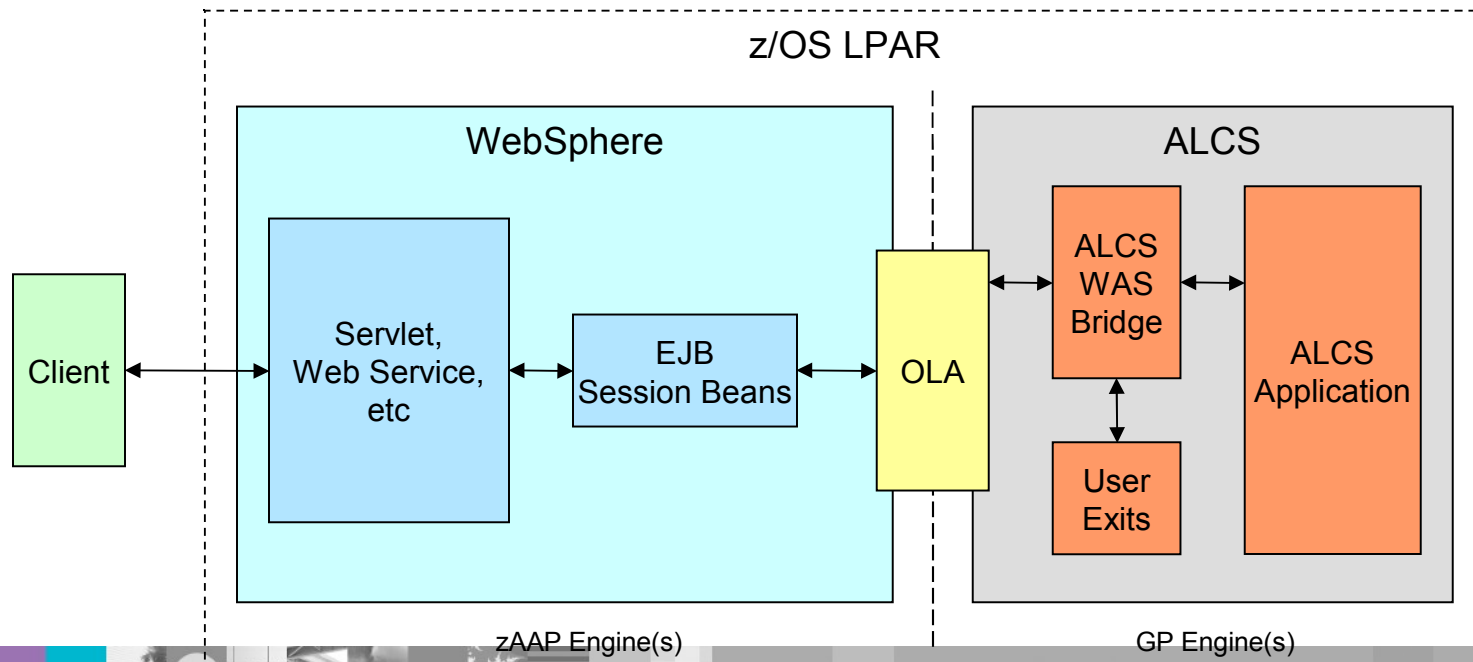
## ALCS and WAS – OLA

- ALCS is a fast transaction processing facility. But ALCS does not try to be all things to all people
- WAS is a very rich open standard
- OLA connectivity is cross memory and is very fast
  - ▶ Performance
  - ▶ Calls are spread through the control region to available WAS servants improving availability and scalability
- **ALCS: The ALCS WAS bridge and ALCS applications**

# ALCS Web Services: WebSphere Optimized Local Adapter

IBM introduced the *WebSphere Optimized Local Adapter (OLA)* as part of WebSphere Application Server for z/OS. The OLA allows a z/OS application, such as ALCS, to communicate with WebSphere Application Server for z/OS J2EE applications.

- ✓ Extremely efficient communication by using shared memory services
- ✓ WebSphere Application Server can utilize cost efficient zAAP engines
- ✓ OLA uses standard J2EE programming paradigms
- ✓ Co-location of WebSphere Application Server and ALCS provides superior reliability



# ALCS and OLA – A straightforward implementation

- ▶ Simple and straightforward
- ▶ New Monitor CSECT called DXCWAS containing following routines
  - Application program support
  - ZCWAS command support
  - Monitor support
- ▶ New Monitor CSECT called DXCWSB containing following routines
  - WAS bridge initialization
  - Start / stop WAS bridge communication
  - WAS bridge receive routine
  - WAS bridge send routine



# Prerequisites

## ALCS

The necessary Optimized Local Adapter support is contained in the WAS Bridge module which is part of APAR PK83249 and requires ALCS version 2.4.1. In addition APAR PK86999 contains the sample skeleton user exits for the WAS Bridge.

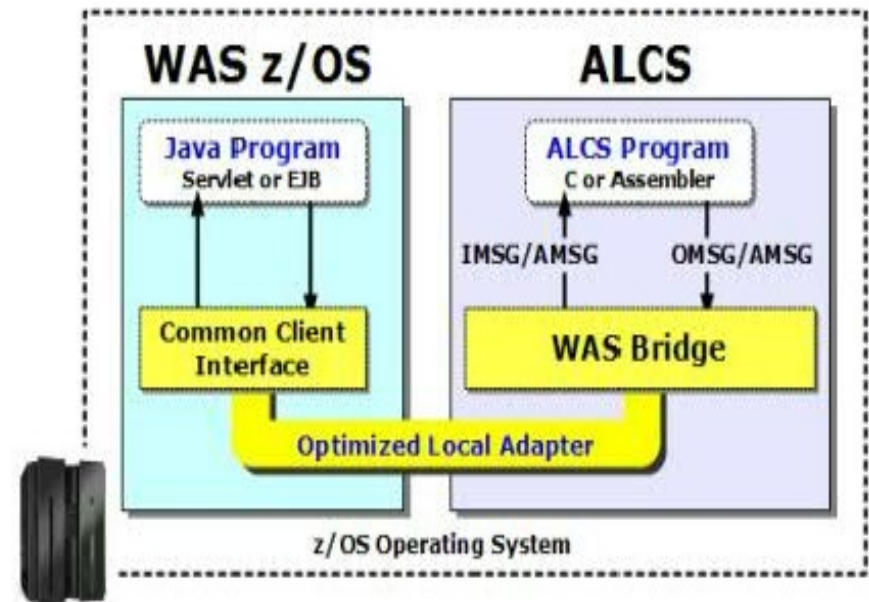
## WebSphere

The Optimized Local Adapter is part of APAR PK85842 which is included in Service Level 7.0.0.4 of WebSphere Application Server for z/OS.



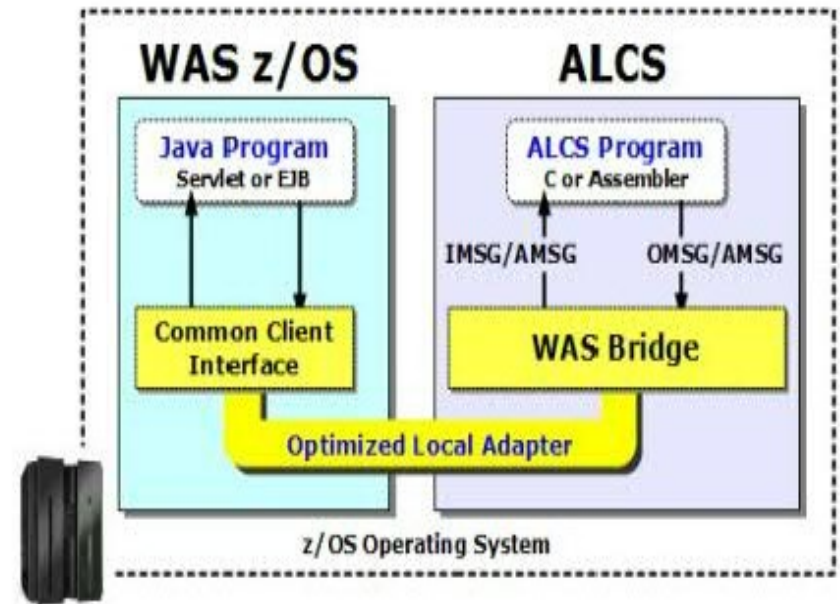
# ALCS WAS bridge

- **ALCS WAS BRIDGE AND OPTIMIZED LOCAL ADAPTERS**
- The ALCS WAS Bridge provides connectivity between current ALCS applications and WebSphere applications. It allows input messages to be formatted and passed on to legacy applications as if they came from ordinary devices. Output ALCS application messages can also be routed to the WAS Bridge.
- Existing ALCS applications can continue to use the traditional ALCS APIs by taking advantage of the ALCS WAS Bridge.



## ALCS WAS bridge (cont.)

The ALCS WAS bridge is designed to minimize any overhead as much as possible. With help of OLA the processing will mainly take place on the Websphere zAAPs. The messages will not be re-formatted neither by ALCS nor by the WAS bridge.



# ALCS WAS bridge

- Documentation is available
  - ▶ Correlators
  - ▶ Message formats
  - ▶ Installation-wide exits
- Useful sample code is available
- ALCS web site
- **A few examples are shown in the following foils**



Select CRI

ALCS Command

```
15MAR
1 TPE LAX 14-1600 14-1120 F7C7Y7 CI 006 744 0
2 TPE LAX 14-1815 14-1335 F7C7Y7 BR 012 744 0
3 TPE LAX 15-1600 15-1120 F7C7Y7 CI 006 744 0
4 TPE LAX 15-1815 15-1335 F7C7Y7 BR 012 744 0
>
```

```

2f d8 31 35 4d 41 52 20 0a 31 20 54 50 45 20 4c      /.15MAR .1 TPE L
41 58 20 31 34 2d 31 36 30 30 20 31 34 2d 31 31      AX 14-1600 14-11
32 30 20 20 46 37 43 37 59 37 20 43 49 20 30 30      20 F7C7Y7 CI 00
36 20 37 34 34 20 20 20 20 20 30 20 0a 32 20 54      6 744 0 .2 T
50 45 20 4c 41 58 20 31 34 2d 31 38 31 35 20 31      PE LAX 14-1815 1
34 2d 31 33 33 35 20 20 46 37 43 37 59 37 20 42      4-1335 F7C7Y7 B
52 20 30 31 32 20 37 34 34 20 20 20 20 20 30 20      R 012 744 0
0a 33 20 54 50 45 20 4c 41 58 20 31 35 2d 31 36      .3 TPE LAX 15-16
30 30 20 31 35 2d 31 31 32 30 20 20 46 37 43 37      00 15-1120 F7C7
59 37 20 43 49 20 30 30 36 20 37 34 34 20 20 20      Y7 CI 006 744
20 20 30 20 0a 34 20 54 50 45 20 4c 41 58 20 31      0 .4 TPE LAX 1
35 2d 31 38 31 35 20 31 35 2d 31 33 33 35 20 20      5-1815 15-1335
46 37 43 37 59 37 20 42 52 20 30 31 32 20 37 34      F7C7Y7 BR 012 74
34 20 20 20 20 20 30 20 0a 3e                        4 0 .>
```

# OTA Air Availability Message (using XML/SOAP)

- ▶ Accepts as input an OTA Air Availability Message (OTA\_AirAvailRQ)
- ▶ Parses the OTA input Message and builds an IPARS Availability Request.
- ▶ Sends the IPARS Availability Request to a target ALCS system.
- ▶ Parses the IPARS Availability Response and builds an OTA Air Availability Response (OTA\_AirAvailRS) which is then returned to the caller
- ▶ Sample code

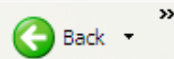


```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <soapenv:Header />
- <soapenv:Body>
- <GetAvailability xmlns="http://www.example.org/AirAvailService/">
  - <OTA_AirAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_AirAvailRQ.xsd" EchoToken="12345"
    Timestamp="2011-04-13T09:30:47-05:00" Target="Production" Version="2.001" SequenceNmbr="1" PrimaryLangID="en-us"
    MaxResponses="10" DirectFlightsOnly="false">
  - <POS>
    <Source AgentDutyCode="BSIA0100AB/SU" />
  </POS>
  - <OriginDestinationInformation>
    <DepartureDateTime>2011-03-15T12:00:00</DepartureDateTime>
    <OriginLocation LocationCode="TPE" />
    <DestinationLocation LocationCode="LAX" />
  </OriginDestinationInformation>
  </OTA_AirAvailRQ>
</GetAvailability>
</soapenv:Body>
</soapenv:Envelope>
```

```

<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
- <soapenv:Body>
- <ns2:GetAvailabilityResponse xmlns="http://www.opentravel.org/OTA/2003/05"
  xmlns:ns2="http://www.example.org/AirAvailService/">
- <OTA_AirAvailRS PrimaryLangID="us-en" Target="Sample" TimeStamp="2011-04-09T15:10:48.018-00:05">
- <OriginDestinationInformation>
  <DepartureDateTime>2011-03-14T16:00</DepartureDateTime>
  <ArrivalDateTime>2011-03-14T11:20</ArrivalDateTime>
  <OriginLocation LocationCode="TPE" />
  <DestinationLocation LocationCode="LAX" />
- <OriginDestinationOptions>
- <OriginDestinationOption>
- <FlightSegment FlightNumber="006" StopQuantity="0">
  <OperatingAirline Code="CI" />
  <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="F" />
  <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="C" />
  <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="Y" />
  </FlightSegment>
  </OriginDestinationOption>
</OriginDestinationOptions>
</OriginDestinationInformation>
- <OriginDestinationInformation>
  <DepartureDateTime>2011-03-14T18:15</DepartureDateTime>
  <ArrivalDateTime>2011-03-14T13:35</ArrivalDateTime>
  <OriginLocation LocationCode="TPE" />
  <DestinationLocation LocationCode="LAX" />
- <OriginDestinationOptions>
- <OriginDestinationOption>
- <FlightSegment FlightNumber="012" StopQuantity="0">
  <OperatingAirline Code="BR" />
  <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="F" />
  <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="C" />
  <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="Y" />
  </FlightSegment>
  </OriginDestinationOption>
</OriginDestinationOptions>

```



```

</OriginDestinationInformation>
- <OriginDestinationInformation>
  <DepartureDateTime>2011-03-15T16:00</DepartureDateTime>
  <ArrivalDateTime>2011-03-15T11:20</ArrivalDateTime>
  <OriginLocation LocationCode="TPE" />
  <DestinationLocation LocationCode="LAX" />
- <OriginDestinationOptions>
  - <OriginDestinationOption>
    - <FlightSegment FlightNumber="006" StopQuantity="0">
      <OperatingAirline Code="CI" />
      <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="F" />
      <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="C" />
      <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="Y" />
    </FlightSegment>
  </OriginDestinationOption>
</OriginDestinationOptions>
</OriginDestinationInformation>
- <OriginDestinationInformation>
  <DepartureDateTime>2011-03-15T18:15</DepartureDateTime>
  <ArrivalDateTime>2011-03-15T13:35</ArrivalDateTime>
  <OriginLocation LocationCode="TPE" />
  <DestinationLocation LocationCode="LAX" />
- <OriginDestinationOptions>
  - <OriginDestinationOption>
    - <FlightSegment FlightNumber="012" StopQuantity="0">
      <OperatingAirline Code="BR" />
      <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="F" />
      <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="C" />
      <BookingClassAvail ResBookDesigQuantity="7" ResBookDesigCode="Y" />
    </FlightSegment>
  </OriginDestinationOption>
</OriginDestinationOptions>
</OriginDestinationInformation>
</OTA_AirAvailRS>
</ns2:GetAvailabilityResponse>
</soapenv:Body>
</soapenv:Envelope>

```

# ALCS ECB-controlled programs

- You can code OLA callable services in assembler and C/C++ application programs.
- WAS calling ALCS applications is also supported
- Here is an example in an assembler ECB-controlled application program:

```
CALL BBOA1REG,(Daemon_Name,  
              Node_Name,  
              Server_Name,  
              Register_Name,  
              Minconn,  
              Maxconn,  
              Flags,  
              Rc,  
              Rsn),VL,MF=(E,APIPlist)
```

# ALCS ECB controlled programs

- Documentation is available
- Sample code is available
- ALCS web site



## Some technical information – ALCS generation

- **GENERATING THE SYSTEM CONFIGURATION TABLE**
- There is a new, optional WAS parameter on the ALCS system generation SCTGEN macro. When you require OLA then you must change your ALCS generation.
- **GENERATING A COMMUNICATION LOAD MODULE WHEN USING THE WAS BRIDGE**
- There are two new COMDEF LDTYPE values for the WAS Bridge communication resources (LDTYPE=WAS and LDTYPE=WASTERM).
- OCTM supports WASTERM.



## Some technical information – Exits

- **ALCS INSTALLATION-WIDE MONITOR EXITS**
- WAS authorization exit — USRWAS1
- ALCS calls this exit before it processes any OLA callable services ( BBOA1xxx) made by ALCS ECB-controlled applications.
- Use this exit to validate or restrict (or both) authorization for the OLA call, according the originating terminal address.
- You may also use this exit to identify an "input" or "output" message. This causes the count of input or output WAS messages to be incremented. ZSTAT MSG or ZSTAT ALL displays these counts. If data collection is currently collecting statistics about input and output messages, this also causes the statistics for input or output WAS messages to be incremented.
- NB. USRWAS2 is reserved for future use.

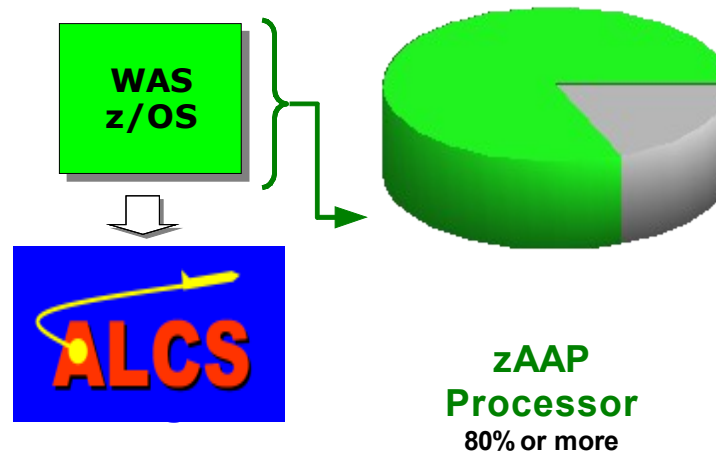


## Some technical information – Exits

- WAS input bridge address exit – USRWAS3
- Use this exit to identify the originating terminal. By default, the WAS bridge assumes there is no originating terminal; it uses the WAS resource as the originator instead. Use this exit when a correlator is used to identify the originating terminal.
- WAS output bridge address exit — USRWAS5
- Use this exit to add a correlator which identifies the destination terminal. ALCS will add the correlator in front of the message on return from this exit.
- NB. By default, the WAS output bridge assumes that the message does not require any reformatting. Use exits USRWAS4 and USRWAS6 when reformatting is required.

## Some technical information – BENCHMARK TEST

- Benchmark study of WAS z/OS exercising 600 transactions per second to ALCS using OLA yielded a zAAP engine usage of approximately 80% of overall z/OS CPU usage.
- Result: high transaction rates with most CPU cycles offloaded to the lower-cost zAAP for a competitive execution model.



## Work in plan

- IBM intends to:
  - ▶ Define a standard correlator and implement a “session management system”. This eliminates the need for ALCS installation-wide exits and will provide a "non-static" mechanism to "link" ALCS and WAS "resources". Customer unique processing such as sign-in may be done in new installation-wide exits.
  - ▶ Modify the (ALCS) scrolling package to send the complete message over the WAS bridge.
  - ▶ Add more examples, which will use Websphere facilities and make use of XML/SOAP.

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

# Any Questions?

- [alcs@uk.ibm.com](mailto:alcs@uk.ibm.com)

