

Modernizing business critical applications with open languages on z/OS

Day 1

Overview of the open-source languages on z/OS – Go, Node.js, and Python

Speakers



Yves Tolod
IBM Z Client Technical Professional

As a Z veteran, Yves brings unique perspectives gained from working with different clients to unleash the true potential of IBM Z. Using real-world examples, Yves sheds light on how clients can bring the most value out of their IBM Z investments.



Jennifer Rowan
Offering Manager - Python and Node.js on z/OS

Prior to her current position, Jennifer held many positions in Development Support, Security, Project Management and Lab Services. Jennifer is passionate about Open Source and application modernization on IBM Z.



James Tang
Offering Manager – Go and Java on z/OS

James is responsible for Go (or Golang) and Java on z/OS. Prior to his current position, James was an offering manager in the Data and AI organization leading the offering delivery of SaaS products.

Panelists



Rosanne Jolin
Manager and Delivery Manager, Python and Node.js for z/OS



Wayne Zhang
Software Developer, z/OS Node.js



Chad McIntyre
IBM Open Enterprise Python for z/OS Dev Lead



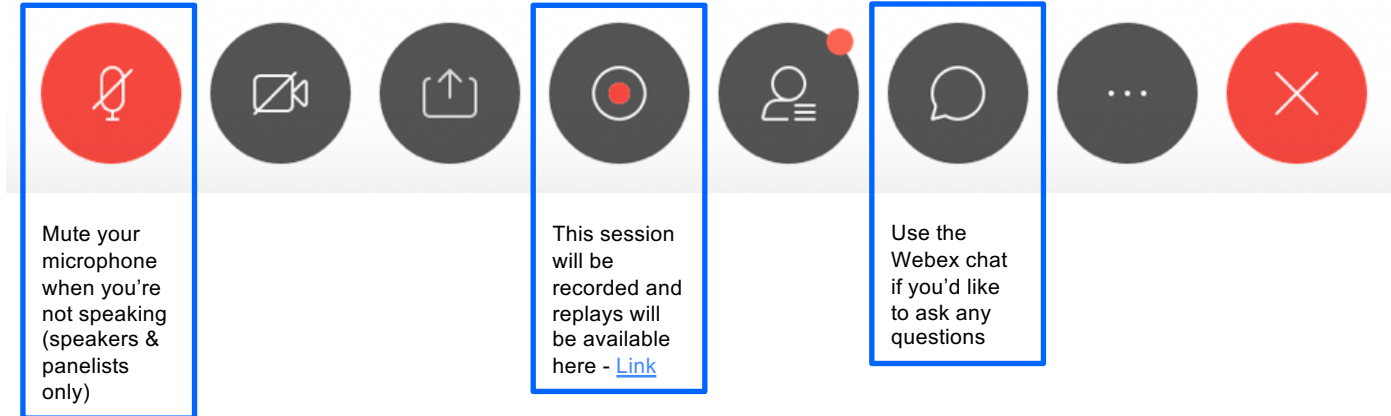
Larry Lindsay
Senior Development Manager and Security Architect



Bill O'Farrell
Team lead for Go Language

Instructions

- If you do not hear audio, check/change your audio connection by clicking either the Phone icon in the Webex controls (if available) or the (...) icon then Switch Audio
- When using chat to ask questions, select To EVERYONE -- the last option in the list. Panelists cannot see chats to ALL ATTENDEES or HOST
- Our technical development team is standing by and ready to answer your questions through chat
- The slides for this presentation and some other resources are available for download in the chat window
- We encourage you to participate in our polling questions as it would help us improve the content for our future events



The challenge

The number of new applications and the cost of maintaining existing applications is skyrocketing

It's important for clients to embark on a strategy to modernize applications and build new one because:

- **Modern applications enable speed** to develop and deploy new capabilities, responding to market demands with the tools to compete and win
- **Modern applications aren't challenged** with technical debt and skills shortages of older technology
- **Modern applications require** that the developer experience change from working on monolithic, slow and cumbersome apps to a fast, agile and responsive culture
- **Modern applications are more secure** and meet compliance standards

500m

New Apps from 2018 to 2025 ¹

40%

Technical debt²

1. Source: IDC FutureScape: Worldwide IT Industry 2020 Predictions October, 2019
2. Source: Gartner: Application Modernization Should Be Business-Centric, Continuous and Multiplatform, August, 2019

IBM's approach to building and modernizing applications maximizes agility and ROI

Strategy & roadmap

Optimize core apps

Application development

Access to data and services

Managing at scale

Culture, people, skills



IBM z15™ is most advanced business platform on the planet

It doesn't need to be modernized; you need to modernize the way you use it



Cloud native

container based
development
and deployment



Encryption everywhere

protect eligible data across
hybrid cloud environments



Cyber resiliency

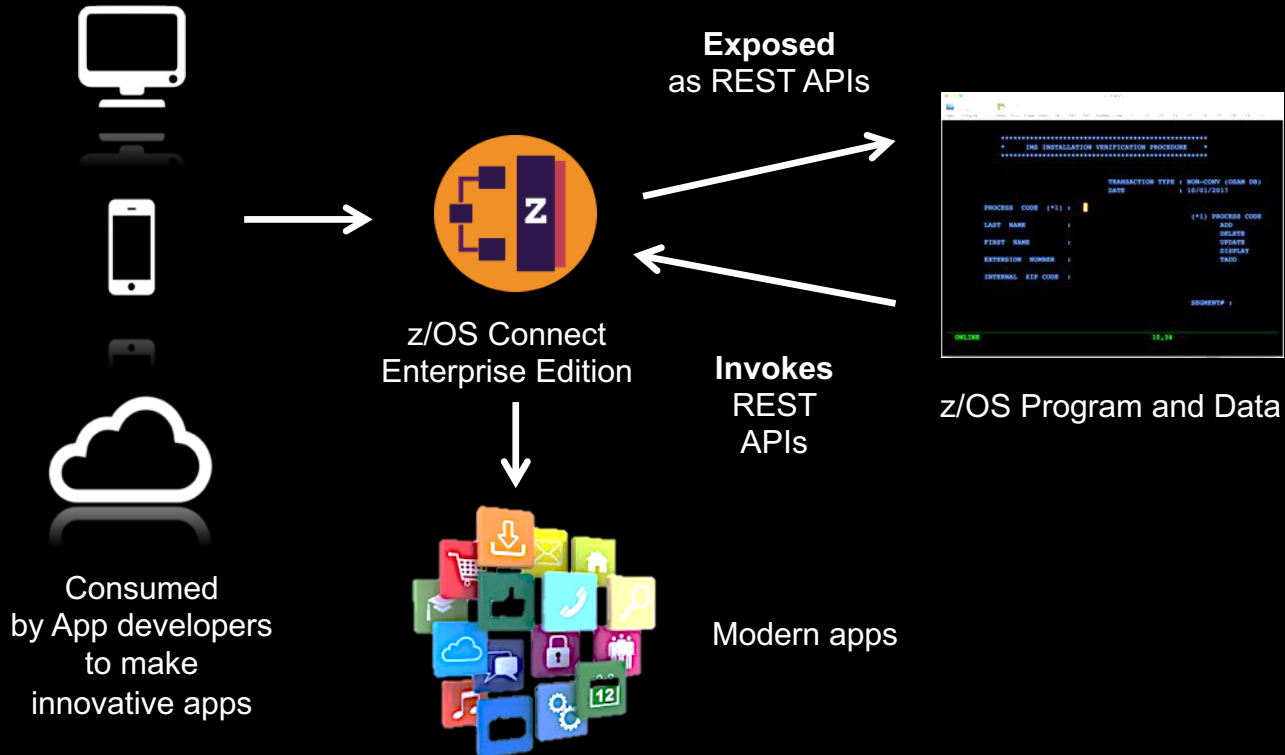
continuous availability
and scalable isolation
of workloads

Designed for flexible computing

new option for high availability plus accelerators

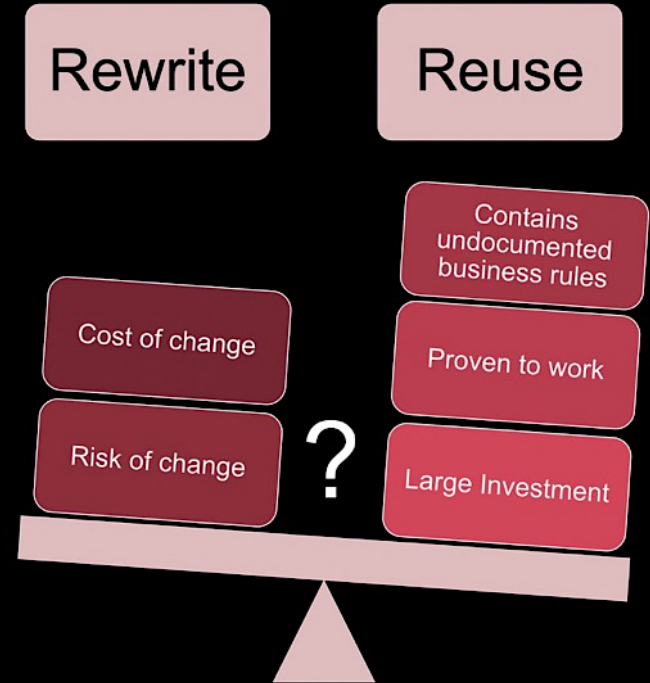
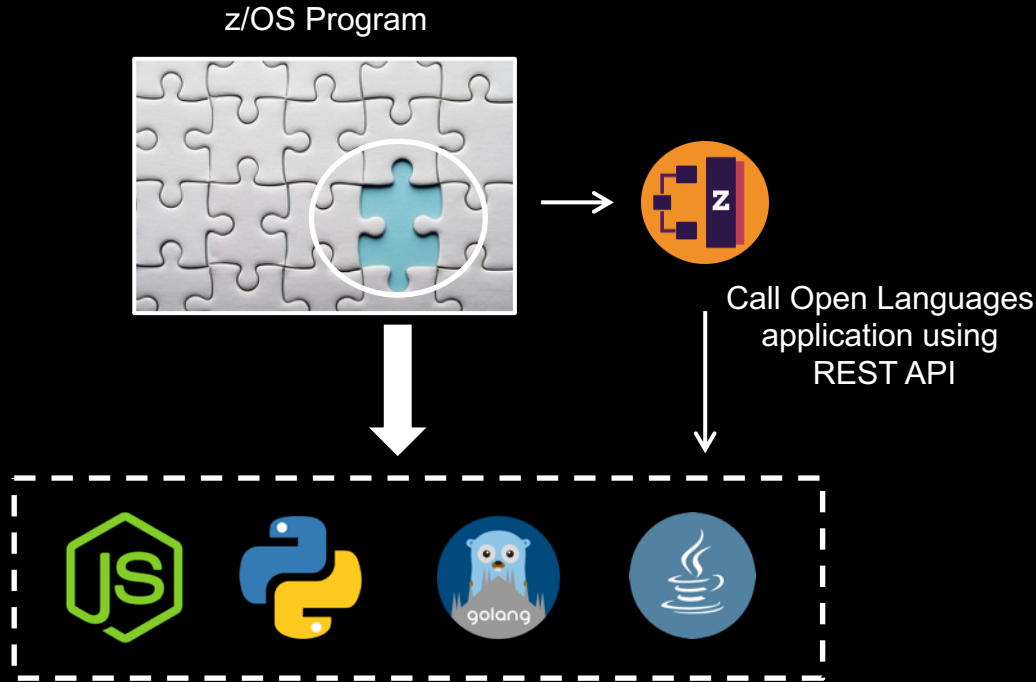
Modern applications are fueled by **APIs**

Integration of Z assets via APIs is seen as the most effective first step on a hybrid cloud journey



Extend existing Core Applications on z/OS with **Open Languages**

Invoked as APIs to accelerate modernization



Why modernize with open programming languages

- Access a vast pool of developers to work on z/OS
- Leverage a large number of “good-to-go” packages to use instantly in applications
- Accelerate digital transformation and enable new applications on IBM Z
- Co-locate applications and services on z/OS to improve performance in accessing critical applications and data



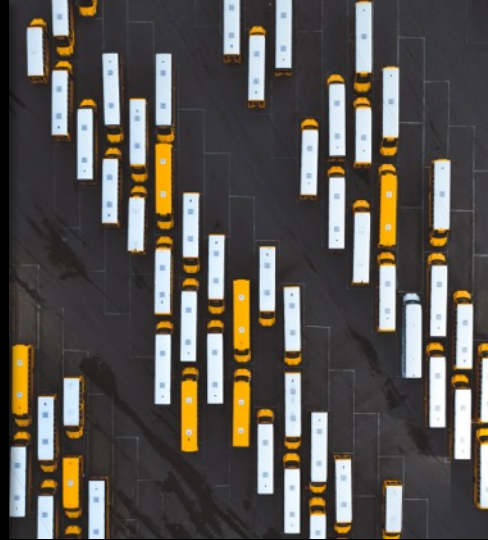
Estimated number of developers worldwide (2021)¹

Python: 8.4M

Node.js: 7.5M

Go: 5.6M

¹IDC - PaaSView and the Developer 2020: Focus Topic Pivot Table — Worldwide and Regional



Support IBM offerings and latest technology development

- **Python:** Red Hat Ansible Certified Content for IBM Z, ZOAU
- **Node.js:** Wazi, Zowe
- **Go:** Container runtime and Kubernetes orchestration for z/OS
SOD: [Link](#)

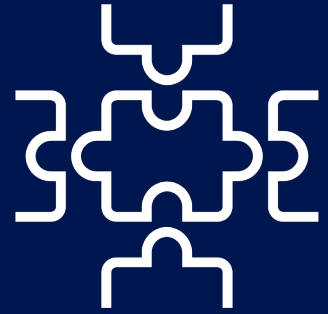
Now available in Shopz

- **Zero license charge**
- **Optional priced S&S**

IBM Open Enterprise SDK for Node.js

What is the IBM Open Enterprise SDK for Node.js - z/OS Offering?

Run Node.js
applications in
z/OS UNIX shell



What is Node.js?

- Server-side JavaScript runtime
- Designed to build scalable network applications
- Best suited for data-intensive (i.e. I/O bound) applications

IBM Open Enterprise SDK for Node.js is a \$0 license charge product with optional IBM Subscription & Support

npmjs.org

Repository of community contributed packages

Over 1.5M packages covering all domains

IBM Open Enterprise SDK for Node.js on z/OS use cases

Digital Transformation

Modernize applications by extending existing applications using Node.js – z/OS microservices on Z

API Orchestration

Avoid exposing more APIs than necessary, simplify API development and maintenance

Interoperability

Access z/OS based resources from Node.js using REST APIs

Application Development

Develop web Applications and create REST APIs

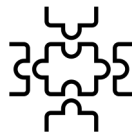
Benefits



Replace monolithic design



Improved organizational efficiency



Better performance. Better security



Modernize in place. Ability to write modern interfaces on z/OS

Why IBM Open Enterprise SDK for Node.js on z/OS?

- JavaScript is **ubiquitous**, available everywhere
- Large open-source community support & largest repository of modules
- **Co-location** of Node.js applications with **critical assets** (i.e. applications and data) on z/OS to reduce response time, increase throughput and increase security
- Directly connect your Node.js applications to your critical assets with Z native modules
- Drive digital transformation



Learn more about Node.js

Product page:

IBM Open Enterprise SDK for Node.js - [Link](#)

Node.js Blog Community:

Latest information, how-to and more - [Link](#)

Technical information:

Visit IBM Documentation - [Link](#)

Additional information:

Node.js.org - [Link](#)

Quick Poll

IBM Open Enterprise SDK for Python

Product Overview

IBM Open Enterprise SDK for Python is based on the popular Python interpreter from Python Software Foundation (PSF)

Current Releases:

IBM Open Enterprise SDK
Python 3.8

IBM Open Enterprise SDK
Python 3.9

Pricing

\$0 license charge
product with
optional S&S

Application: 64-bit, ASCII/XPLINK

Packages shipped with the
SDK include:

Python stdlib, cffi, cryptography, numpy, pycparser,
pip, setuptools, six, zos_util

Python Package Index (PyPI)

Repository of community contributed packages
Over 200,000 packages

Packages cover all domains, web hosting, scientific
computing, databases, etc.

IBM Open Enterprise SDK for Python use cases

Business Transformation

Z clients can extend core business critical applications using Python

Data Science

Perform advanced data analysis using the most popular python packages on the platform where the data is stored

Interact with z/OS

Using Z Open Automation Utilities, you can utilize the existing infrastructure and interact with different utilities (datasets, JCL, etc.)

Ansible via Python

Automate applications and IT infrastructure using Python via Ansible

Benefits



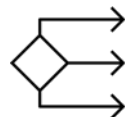
Drive business transformation. Modernize in place



Co-locating the data and Python applications on Z will improve performance and keep data secure



Simplify. Automate. Don't need to move your data



Manage your z/OS resources the same way as you would on distributed

Why IBM Open Enterprise SDK for Python on z/OS?

- Develop Python applications on Z the same way you would on any other platform
- Leverage a large ecosystem of packages and frameworks to shorten delivery time
- Drive digital transformation
- Co-locating the data and Python applications on Z will improve performance and keep data secure
- Python on z/OS enables Z clients to use the de facto language to develop Red Hat® Ansible modules to streamline automation operations for their hybrid IT environment



Learn more about Python

Product page: IBM Open Enterprise SDK for Python - [Link](#)

Python Blog Community: Latest information, how-to and more - [Link](#)

Technical information: Visit IBM Documentation - [Link](#)

Additional information: Python.org - [Link](#)

Quick Poll

IBM Open Enterprise SDK for Go

Product Overview

IBM Open Enterprise SDK for Go is an industry-standard Go (or Golang) compiler that brings a powerful framework for building fast and scalable applications to the z/OS platform.

Current Release:

IBM Open Enterprise SDK
for Go 1.16
(GA'ed March 12, 2021)

Pricing

\$0 license charge
product with optional
S&S

Technical details

Software requirements

- z/OS V2R4, or later
- IBM z/OS UNIX System Services

Hardware requirements

- IBM Z servers that support z/OS V2R4, or later
- Required hard disk space is minimum 850 MB

Full enablement of Go on z/OS

- Go standard runtime library
- UTF-8 Unicode Standard
- Go modules and multithreading
- Enablement of Go to call C code

IBM Open Enterprise SDK for Go use cases

Modernization in place

Develop APIs / plugins / wrappers to extend business critical applications using Go for faster delivery, ease of maintenance and lower cost

Enable popular Go applications on z/OS

Applications written in Go on another platform can run more efficiently, securely and with higher performance on z/OS (Kubernetes, Open Container Initiative, Analytics, AI, etc)

Interoperate with middleware / other languages

Communicate and integrate with existing middleware via z/OS Connect APIs and other languages via cgo

Develop new applications

Build applications that are more lightweight, use less system resources, and perform better overall

Benefits



Simplicity - Simple syntax, lightweight, easy to understand & maintain



Agility - Microservice architecture, multiple manageable services



Performance - Fast running due to small syntax and concurrency support, highly scalable



Ecosystem - Access to millions of Go developers worldwide

Why IBM Open Enterprise SDK for Go on z/OS?

- Leverage colocation of Go applications with critical assets and data on z/OS, for faster response times and more throughput
- Run popular Go applications from other platforms on Z, for greater reliability and resiliency
- Take advantage of great features supported by a large open community, for greater agility
- Leverage Go's powerful standard library of packages and built-in functions, for rapid development



Learn more about Go

Product page: IBM Open Enterprise SDK for Go - [Link](#)

Go Blog Community: Latest information, how-to and more - [Link](#)

Technical information: Visit IBM Documentation - [Link](#)

Additional information: golang.org - [Link](#)

Quick Poll

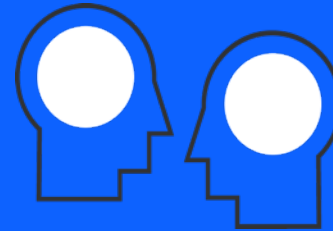
Q&A

The IBM Z Sponsor User Program helps address your business priorities and solve your business problems

- Collaborate with IBM Designers & Technical Leaders to solve customer challenges
- Influence product design and direction through the IBM Design Thinking process
- Engage with the IBM Development teams
- Gain visibility to product requirements, design, development and test progress
- Understand details of technical previews for collaborative activities
- Open to all* Customers and Business Partners

IBM Sponsor User Program

- All design feedback activities are covered by existing DPP Agreement you will all have signed.
- Allows IBM to share roadmaps, designs, use cases and other confidential materials.



- Discuss your organization's needs/challenges and help us shape the product's future.
- There is no fee to participate in this program.

Contact Jennifer Rowan: jenrow@ca.ibm.com

Thank you for joining us today and don't forget to attend tomorrow's deep-dive session.

Here's what you can expect in tomorrow's session –

- How to create a simple RESTful backend using Go internal HTTP server: Use battery-included features in Go to create standalone web servers to parse JSON format.
- How to extend core business applications with Node.js: The first demo shows how you can extend an existing CICS and IMS COBOL application by calling a microservice written in Node.js. The second demo shows how can you create a Backend For Frontend (BFF) layer with Node.js to make multiple calls - a call to CICS and a call to Db2 - and expose it as a single call.
- Exploring Binary classification with Python. This demo will show you how to perform exploratory analysis on your existing data, normalize values, generate plots and figures (heat maps, histograms, etc.), and prepare the data for classification. Execute binary classification algorithm on the pre-processed data and classify new data according to the trained model.

Replay link will be emailed soon.

THANK YOU

謝謝

DZIĘKUJĘ CI

NGIYABONGA

TEŞEKKÜR EDERİM

DANKIE

TERIMA KASIH

СПАСИБО

GRAZIE

PAKMET CI3GE

GO RAIBH MAITH AGAT

БЛАГОДАРЯ

GRACIAS

ТИ БЛАГОДАРАМ

TAK DANKE

RAHMAT

MERCI

HATUR NUHUN

PAXMAT CAҒA

CẨM ƠN BẠN

WAZVIITA

TAPADH LEIBH

KEA LEBOHA

MISAOTRA ANAO

WHAKAWHETAI KOE

DANKON

TANK

TAPADH LEAT

MATUR NUWUN

ХВАЛА ВАМ

MULTUMESC

고맙습니다

GRAZIE

شكرا

FAAFETAI

ESKERRIK ASKO

HVALA

TEŞEKKÜR EDERİM

OBRIGADO

MERCI

DI OU MÈSI

ĐAKUJEM

SIPAS JI WERE

TERIMA KASIH

UA TSAUG RAU KOJ

ТИ БЛАГОДАРАМ

СИПОС

KÖSZÖNÖM

GRÁCIÉS

MAHADSANID

DANK JE

EΥΧΑΡΙΣΤΩ

GRATIAS TIBI

AČIŮ

SALAMAT

MAHALO IĀ 'ŌE

TAKK SKAL DU HA

GRAZZI

PAKKA PĒR

FALEMINDERIT

ДЗЯКУЮ

Notices and disclaimers

© 2021 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers

continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and 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.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

IBM