Mining Gold from RMF Monitor III
The XML Batch Reporting Facility

Peter Muench (pmuench@de.ibm.com)

IBM Corporation

Monday, March 2, 2015
Session 16817

#SHAREorg

SHARE is an independent volunteer-run information technology association
that provides **education**, professional **networking** and industry **influence**.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Objectives

► Generate RMF Monitor III Reports automatically

► Provide Sysplex wide Reporting Scope

► Store the Reports persistently for each Mintime

► Create an Archive for selected Reports

► Process individual Reports and apply intelligent Analysis

⇨ Convert the Report XML Document to alternate Formats (CSV, JSON)

⇨ Parse the Report XML Document and extract Key Metrics

► Provide a State of the Art Reporting GUI
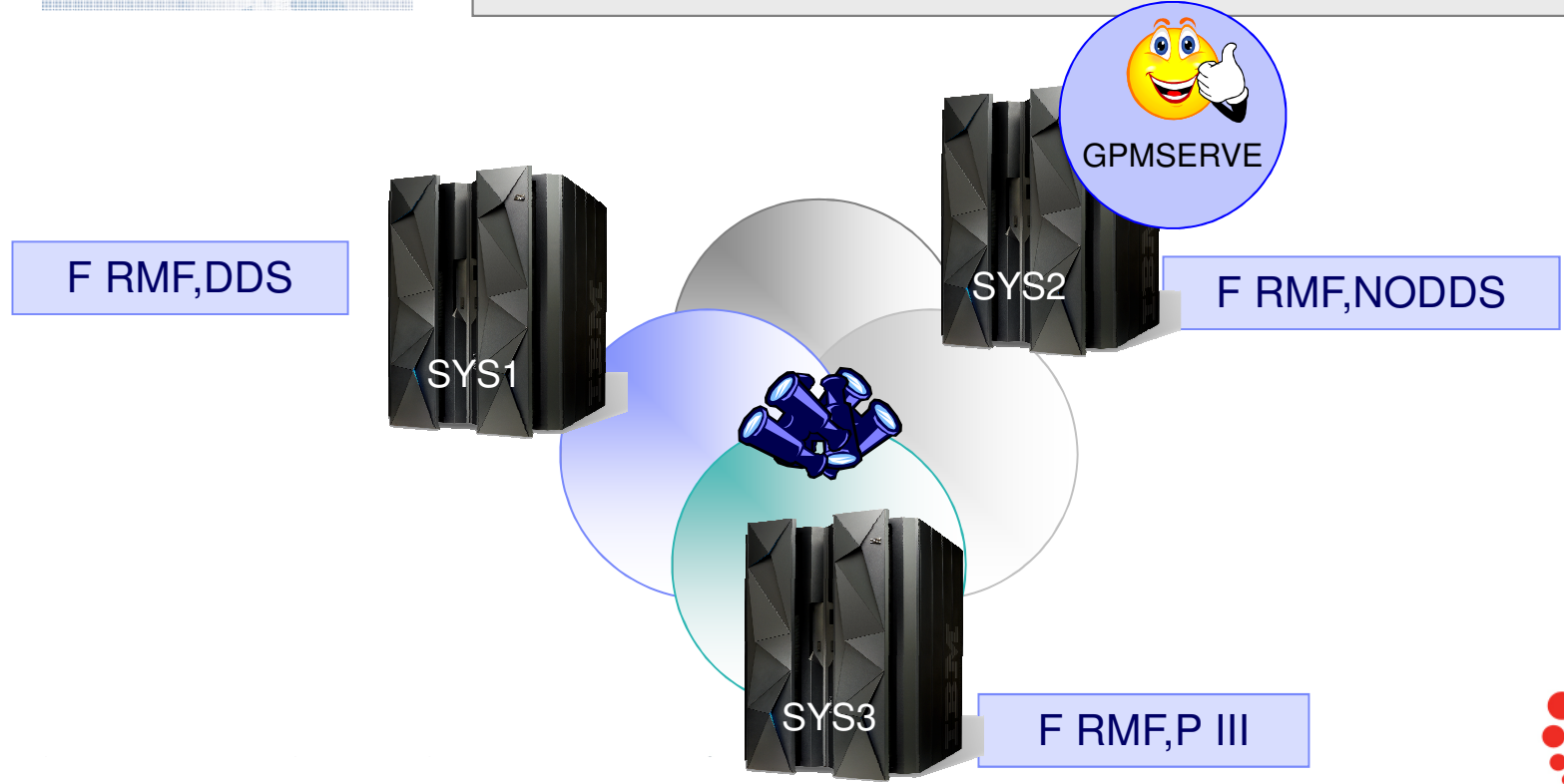
► Avoid cumbersome Downloads to the Workstation

The Monitor III XML Batch Facility (1)

Automatic DDS Detection

PGM=GPMXMLM3

- ► The RMF Distributed Dataserver can generate Monitor III XML Reports
- ► DDS can change Location based on RMF Master System
- ► Module GPMXMLM3 can detect the DDS Location automatically

GPMSERVE

F RMF,DDS

SYS2   F RMF,NODDS

SYS1

SYS3   F RMF,P III

The Monitor III XML Batch Facility (2)

Process Flow

- ► Module GPMXMLM3 is invoked using a standard batch job
- ► XML reports are retrieved by means of HTTP Requests

HTTP Request

http://ddshost:8803/gpm/reports/CPC?resource=,SYS3,MVS_IMAGE

PGM=GPMXMLM3

DDS

SYS1   SYS2

SYS3   SYS4

z/OS Sysplex
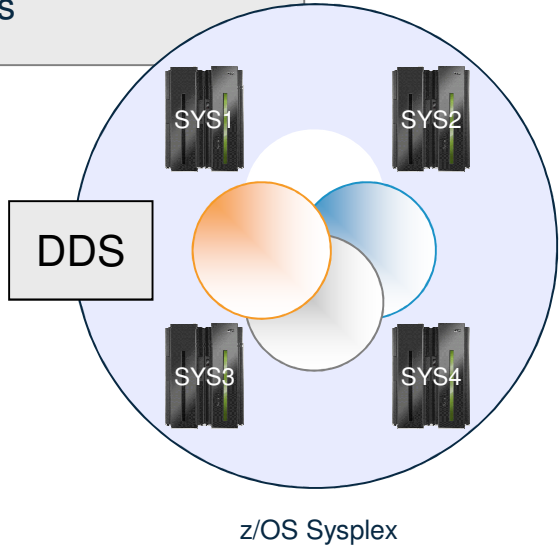
```
<?xml version="1.0">
<ddsml>
  <report>
    <metric id="CPC">
    </metric>
    <caption>
      <var>
        <name>CPCHPNAM</name>
        <value>SYS3</value>
      </var>
      <var>
        <name>CPCHMOD</name>
        <value>2097</value>
      </var>
    </caption>
  </report>
</ddsml>
```

XML Response Document

SHARE in Seattle 2015

The Monitor III XML Batch Facility (3)

Automatic Sysplex Expansion

► Sysplex wide Reports: One HTTP request against the Sysplex resource
► Reports with System scope: One HTTP request per System

PGM=GPMXMLM3

http://ddshost:8803/gpm/reports/CFACT?resource=,ZOS1,SYSPLEX

DDS
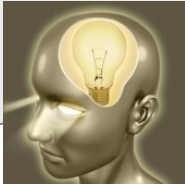
http://ddshost:8803/gpm/reports/CPC?resource=,SYS1,MVS_IMAGE
http://ddshost:8803/gpm/reports/CPC?resource=,SYS2,MVS_IMAGE
http://ddshost:8803/gpm/reports/CPC?resource=,SYS3,MVS_IMAGE
http://ddshost:8803/gpm/reports/CPC?resource=,SYS4,MVS_IMAGE

SYS1   SYS2

SYS3   SYS4
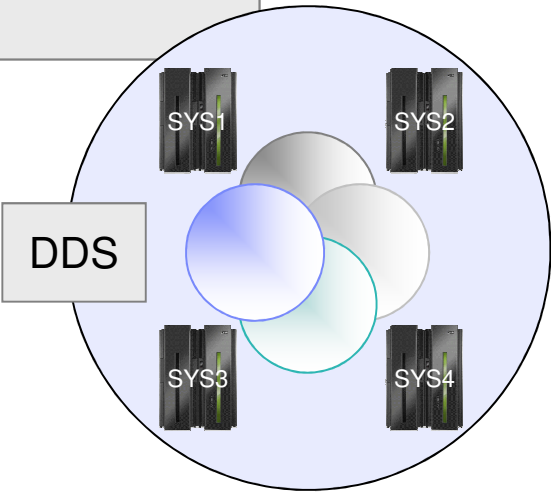
z/OS Sysplex

Advanced feature for reports with System scope:
• query all System names of the Sysplex
• assemble the HTTP requests accordingly
• combine the reports to one XML document

The Monitor III XML Batch Facility (4)

JCL & Module Parameters

```
//M3XML     PROC REPORT=,              //* Report Type                              *//
//               DATE=0,               //* Begin Time/Date                          *//
//               RANGE=0,              //* Length of Reporting Range                *//
//               UID=0,                //* Userid Id (if DDS Login requires Credentials) *//
//               PWD=0,                //* Password (if no Passtickets are configured)   *//
//               APPL=0,               //* Application Name GPMSERVE (for Passtickets)   *//
//               HOST=0,               //* DDS Hostname (if no Autodetection is required) *//
//               PORT=0                //* Port Number (if no default Port 8803)     *//
//*
//GPMM3     EXEC PGM=GPMXMLM3,
//   PARM=('&REPORT &DATE &RANGE &UID &PWD &APPL &HOST &PORT')
//*
//X3RPTS    DD   PATH='/u/rmf/m3xml/temp/&REPORT..xml',   //* USS Output Directory for *//
//               PATHOPTS=(OWRONLY,OCREAT,OTRUNC),        //* Single System Reports    *//
//               PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//X3XSRPTS  DD   PATH= '/u/rmf/m3xml/temp/&REPORT..xml',  //* USS Output Directory for *//
//               PATHOPTS=(OWRONLY,OCREAT,OTRUNC),        //* Sysplex Reports          *//
//               PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//SYSPRINT  DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//          PEND
```
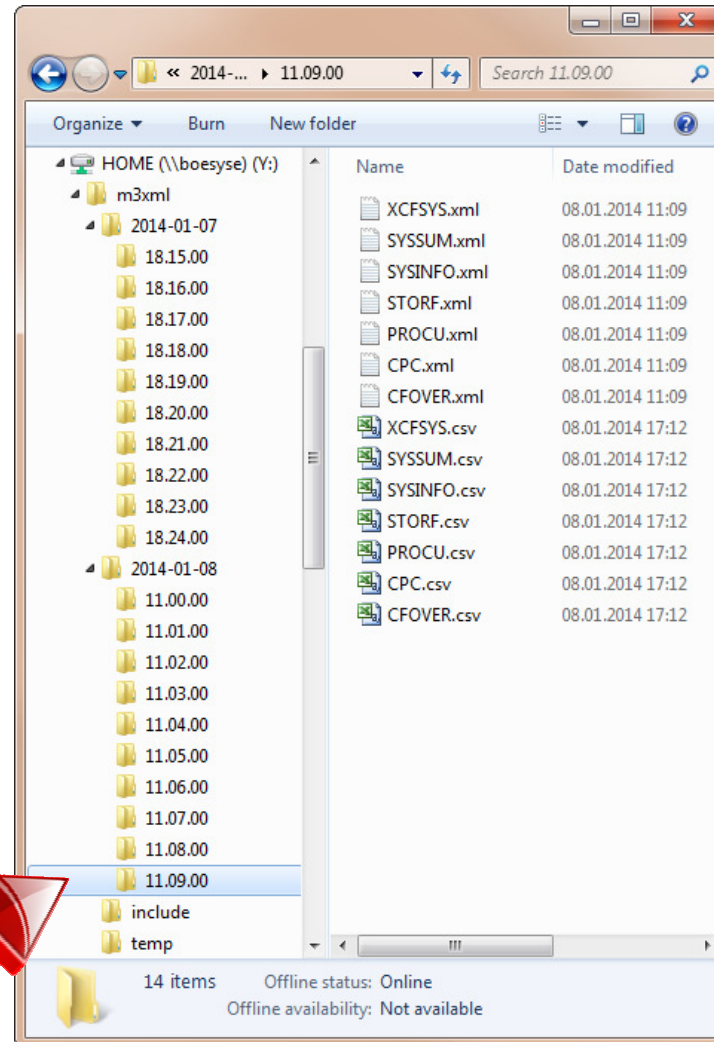
PGM=GPMXMLM3

The Monitor III XML Batch Facility (5)

Security & Passtickets

```
//M3XML     PROC REPORT=,
//               DATE=0,
//               RANGE=0,
//               UID=IBMUSER,
//               PWD=0,
//               APPL=GPMSERVE,
//               HOST=0,
//               PORT=0
```

PGM=GPMXMLM3

This service is invoked on behalf of the userid assigned to the JCL

HTTP Header

IRRSPK00()

The callable service IRRSPK00 returns a passticket for a specific userid and an application name

Base64 encoded string

The passticket is obtained for the userid specified with the UID parameter of the M3XML procedure

RMF Distributed Data Server

__passwd()   RACROUTE REQUEST=VERIFY

RACF
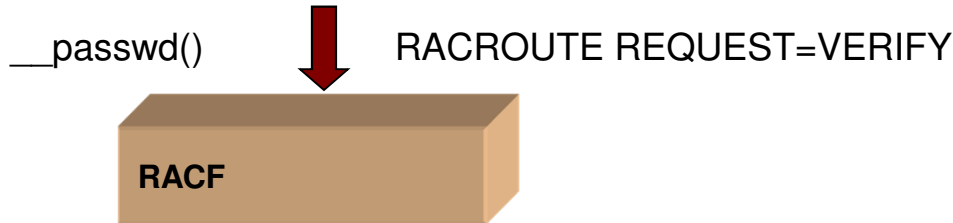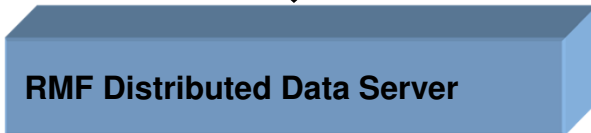
Mintime Scheduled Job Submission

```
//JOBRDR PROC JCL=
//***********************************************************
//*                                                         *
//* NOTES:                                                  *
//*    THIS PROCEDURE SENDS A JOB TO THE INTERNAL READER    *
//*    USAGE: S JOBRDR,JCL= MYDSN(M3XML)                    *
//*                                                         *
//***********************************************************
//READJOB EXEC PGM=IEBGENER
//SYSPRINT DD   DUMMY
//SYSIN    DD   DUMMY
//SYSUT1   DD   DISP=SHR,DSN=&JCL
//SYSUT2   DD   SYSOUT=(A,INTRDR)
```

Looks quite nice to me. But i prefer to use my own job scheduler for this task.

JES2 Automatic Commands

MVS System Commands through JES2

$TA,I=100,'$VS,"S JOBRDR,JCL=MYDSN(M3XML)"'

Monitor III Mintime in Seconds

Charging the Monitor III Report Repository (1)

M3XML Procedure:
Recurrent Job Submission right
after Mintime Completion

```
//M3XML      PROC REPORT=,
//                DATE=0,
//                RANGE=0,
//                UID=0,
//                PWD=0,
//                APPL=0,
//                HOST=0,
//                PORT=0
//*
//GPMM3     EXEC PGM=GPMXMLM3,
//  PARM=('&REPORT &DATE &RANGE &UID
//        PEND
//*
//M3CFO   EXEC M3XML,REPORT=CFOVER
//M3CPC   EXEC M3XML,REPORT=CPC
//M3PRU   EXEC M3XML,REPORT=PROCU
//M3STF   EXEC M3XML,REPORT=STORF
//M3SYSI  EXEC M3XML,REPORT=SYSINFO
//M3SUM   EXEC M3XML,REPORT=SYSSUM
//M3XCFS  EXEC M3XML,REPORT=XCFSYS
//*
```

HOME (\\boesyse.boeblingen.de.ibm.com) (Z:) ▸ m3xml ▸ 2014-01-07 ▸ 18.15.00

Search 18.15.00

Organize ▾   Burn   New folder

HOME (\\boesyse.boeblingen.de.ibm.com) (Z:)
  m3xml
    2014-01-07
      18.15.00
      18.16.00
      18.17.00
      18.18.00
      18.19.00
      18.20.00
      18.21.00
      18.22.00
      18.23.00
      18.24.00
    2014-01-08
      11.00.00
      11.01.00
      11.02.00
      11.03.00
      11.04.00
      11.05.00
      11.06.00
      11.07.00
      11.08.00
      11.09.00
    include
    temp

Name	Date modified	Type	Size
CFOVER.xml	07.01.2014 18:15	XML File	4 KB
CPC.xml	07.01.2014 18:15	XML File	76 KB
PROCU.xml	07.01.2014 18:15	XML File	9 KB
STORF.xml	07.01.2014 18:15	XML File	138 KB
SYSINFO.xml	07.01.2014 18:15	XML File	118 KB
SYSSUM.xml	07.01.2014 18:15	XML File	81 KB
XCFSYS.xml	07.01.2014 18:15	XML File	26 KB

7 items   Status: Online
           bility: Not available

USS Report Repository
mapped by DFS SMB Server

Charging the Monitor III Report Repository (2)

m3xml.sh Shell Script:
Creates Directories based on Report Range
Copies the Reports from /temp Directory

```
//GPMM3      EXEC PGM=GPMXMLM3,
//   PARM=('&REPORT &DATE &RANGE &UID &PWD &APPL &HOST &PORT')
//*
//X3RPTS    DD    PATH='/u/rmf/m3xml/temp/&REPORT..xml',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//X3XSRPTS DD    PATH= '/u/rmf/m3xml/temp/&REPORT..xml' ,
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//*
//COPYXML   EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=0M
//STDPARM  DD *
SH export PATH=$PATH:/u/bhbe/m3xml;
   m3xml.sh /u/bhbe/m3xml CFOVER;
   m3xml.sh /u/bhbe/m3xml CPC;
   m3xml.sh /u/bhbe/m3xml PROCU;
   m3xml.sh /u/bhbe/m3xml STORF;
   m3xml.sh /u/bhbe/m3xml SYSINFO;
   m3xml.sh /u/bhbe/m3xml SYSSUM;
   m3xml.sh /u/bhbe/m3xml XCFSYS;
//
```

Mining Gold – CSV Conversion (1)

```
//M3PARSE  PROC REPORT=,
//              LOW=1,
//              HIGH=9,
//              VAR=,
//              INST=
//*--------------------------------------------------------
//*  GPMXMLPP Parameters:
//*
//*  PARM 1: XML FILENAME
//*  PARM 2: EXTRACTION - LOW BOUND    ==> CSV Extraction
//*  PARM 3: EXTRACTION - HIGH BOUND   ==> CSV Extraction
//*  PARM 4: SEPARATOR FOR CSV FORMAT  ==> CSV Dump
//*  PARM 5: VARIABLE NAME             ==> Table Header Value
//*  PARM 6: INSTANCE NAME             ==> Table Value
//*--------------------------------------------------------
//STEP1     EXEC PGM=GPMXMLPP,
//  PARM=('DD:XMLFILE &LOW &HIGH ; &VAR &INST')
//STEPLIB  DD    DSN=SYS1.SIEALNKE,DISP=SHR
//XMLFILE  DD    PATH='/u/bhbe/m3xml/temp/&REPORT..xml',
//          PATHOPTS=(ORDONLY)
//SYSPRINT DD    PATH='/u/bhbe/m3xml/temp/&REPORT..csv',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//          PEND
//M3CFO     EXEC M3PARSE,REPORT=CFOVER
//M3CPC     EXEC M3PARSE,REPORT=CPC
//M3PRU     EXEC M3PARSE,REPORT=PROCU
//M3STF     EXEC M3PARSE,REPORT=STORF
//M3SYSI    EXEC M3PARSE,REPORT=SYSINFO
//M3SUM     EXEC M3PARSE,REPORT=SYSSUM
//M3XCFS    EXEC M3PARSE,REPORT=XCFSYS
//
```

Module exploits z/OS XML System Services

Mining Gold – CSV Conversion (2)

Mining Gold – Extract Key Metrics (1)

```
//M3PARSE      SYSE      14009 16:14:31.29 BHBE     00000210 D A
//             SYSE      14009 16:14:31.29 BHBE     00000010 IEE114I 16.14.31 2014.009 ACTIVITY 321
//                                           321 00000010 JOBS      M/S      TS USERS    SYSAS    INITS    ACTIVE/MAX VTAM      OAS
//                                           321 00000010 00004     00029    00002       00040    00106    00002/00125          00022
//             SYSE      14009 16:16:45.98 JOB03419 00000211 $HASP100 BHBE$XP  ON INTRDR      BHBE              FROM TSU03418
//                                                           BHBE
//*--------    SYSE      14009 16:16:45.98 JOB03419 00000210 IRR010I  USERID BHBE     IS ASSIGNED TO THIS JOB.
//*   GPMXM    SYSE      14009 16:16:46.02 JOB03419 00000211 ICH70001I BHBE     LAST ACCESS AT 16:11:37 ON THURSDAY, JANUARY 9, 2014
//*            SYSE      14009 16:16:46.03 JOB03419 00000010 $HASP373 BHBE$XW   STARTED - INIT 1    - CLASS A        - SYS SYSE
//*   PARM     SYSE      14009 16:16:46.16 JOB03419 00000010 +XML200I VAR: PRUPCPT  INST: XCFAS  VALUE: 0.6
//*   PARM     SYSE      14009 16:16:46.16 JOB03419 00000010 +XML200I VAR: PRUPCPT  INST: XCFAS  VALUE: 0.9
//*   PARM     SYSE      14009 16:16:46.16 JOB03419 00000010 +XML200I VAR: PRUPCPT  INST: XCFAS  VALUE: 0.6
//*   PARM     SYSE      14009 16:16:46.17 JOB03419 00000210 -                                        --TIMINGS (MINS.)--
//*   PARM     SYSE      14009 16:16:46.17 JOB03419 00000210 -JOBNAME  STEPNAME PROCSTEP    RC    EXCP   CONN     TCB    SRB    CLOCK
//*   PARM                                                    SERV  PG   PAGE   SWAP   VIO SWAPS
//*   PARM     SYSE      14009 16:16:46.17 JOB03419 00000210 -BHBE$XP  STEP1                      00    2259     79 ******    .00      .0
//*   PARM                                                   35615  0    4      0      0     0
//*--------    SYSE      14009 16:16:46.17 JOB03419 00000210 -BHBE$XP  ENDED.  NAME-BHBE              TOTAL TCB CPU TIME=    .00
//                                                           TOTAL ELAPSED TIME=     .0
//STEP1        SYSE      14009 16:16:46.17 JOB03419 00000010 $HASP395 BHBE$XP  ENDED
//   PARM=(    SYSE      14009 16:16:46.18          00000010 $HASP309 INIT 1    INACTIVE ******** C=A
//STEPLIB      SYSE      14009 16:16:46.18 INTERNAL 00000210 SE '16.16.46 JOB03419 $HASP165 BHBE$XP  ENDED AT BOESYS  MAXCC=0000',
//XMLFILE                                                    LOGON,USER=(BHBE)
//           PATHOPTS=(ORDONLY)
//SYSPRINT DD     *
//           PEND
//M3SYSI     EXEC M3PARSE,REPORT=SYSINFO,VAR=SYSLCPVC      //* System wide MVS Utilization % *//
//M3SUM      EXEC M3PARSE,REPORT=SYSSUM,VAR=SUMPFID,INST=BATHI   //* Performance Index          *//
//M3PRU      EXEC M3PARSE,REPORT=PROCU,VAR=PRUPCPT,INST=XCFAS    //* Total Time on CP %         *//
//
```
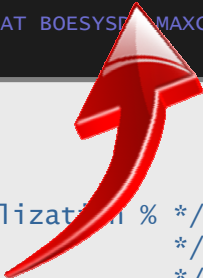
Mining Gold – Extract Key Metrics (2)

```
SYSE      14009 16:16:46.16 JOB03419 00000010  +XML200I VAR: PRUPCPT  INST: XCFAS  VALUE: 0.6
SYSE      14009 16:16:46.16 JOB03419 00000010  +XML200I VAR: PRUPCPT  INST: XCFAS  VALUE: 0.9
SYSE      14009 16:16:46.16 JOB03419 00000010  +XML200I VAR: PRUPCPT  INST: XCFAS  VALUE: 0.6
```

file:///D:/m3xml/2014-01-08/11.09.00/PROCU.xml

RMF Report [,S4,MVS_IMAGE] : PROCU (Processor Usage)

Time Range: 01/08/2014 11:08:00 - 01/08/2014 11:09:00

Jobname	ASID (dec)	Job Class	Job Class Ext	Service Class	Period	Total Time on CP %	AAP Time on CP %	IIP Time on CP %	CP EAppl %	AAP EAppl %	IIP EAppl %	Total Appl %	Total EAppl %	TCB Time %	SRB Time %	P/C SRB %	P/C SRB + Enclave %	Total CPU Time in mSec
XCFAS	0006	S	S	SYSTEM	1	0.6	0.0	0.0	0.6	0.0	0.0	0.6	0.6	0.5	0.1	0.0	0.0	364
RMFGAT	0181	S	SO	SYSSTC	1	0.0	0.0	0.0	0.4	0.0	0.0	0.4	0.4	0.4	0.0	0.0	0.0	256
SMSVSAM	0009	S	S	SYSTEM	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	73

RMF Report [,SYSF,MVS_IMAGE] : PROCU (Processor Usage)

Time Range: 01/08/2014 11:08:00 - 01/08/2014 11:09:00

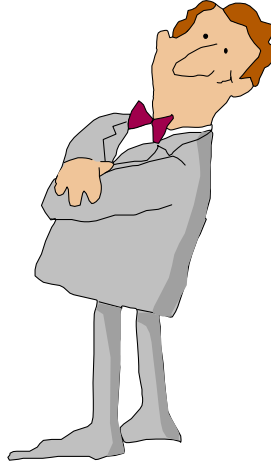| Jobname	ASID (dec)	Job Class	Job Class Ext	Service Class	Period	Total Time on CP %	AAP Time on CP %	IIP Time on CP %	CP EAppl %	AAP EAppl %	IIP EAppl %	Total Appl %	Total EAppl %	TCB Time %	SRB Time %	P/C SRB %	P/C SRB + Enclave %	Total CPU Time in mSec
GPM4CIMX	0116	S	SO	STCDEF	1	1.0	0.0	0.7	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	577
XCFAS	0006	S	S	SYSTEM	1	0.9	0.0	0.0	0.9	0.0	0.0	0.9	0.9	0.8	0.2	0.0	0.0	562
GPM4CIMZ	0117	S	SO	STCDEF	1	0.5	0.0	0.4	0.5	0.0	0.0	0.5	0.5	0.5	0.0	0.0	0.0	276
SMSVSAM	0009	S	S	SYSTEM	1	0.4	0.0	0.0	0.4	0.0	0.0	0.4	0.4	0.4	0.0	0.0	0.0	234
RMFGAT	0066	S	SO	SYSSTC	1	0.4	0.0	0.0	0.4	0.0	0.0	0.4	0.4	0.4	0.0	0.0	0.0	256
TCPIP	0098	S	SO	SYSSTC	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.1	0.0	0.0	88
GPMSERVE	0115	S	SO	GPMSERVE	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	47

RMF Report [,SYSE,MVS_IMAGE] : PROCU (Processor Usage)

Time Range: 01/08/2014 11:08:00 - 01/08/2014 11:09:00

Jobname	ASID (dec)	Job Class	Job Class Ext	Service Class	Period	Total Time on CP %	AAP Time on CP %	IIP Time on CP %	CP EAppl %	AAP EAppl %	IIP EAppl %	Total Appl %	Total EAppl %	TCB Time %	SRB Time %	P/C SRB %	P/C SRB + Enclave %	Total CPU Time in mSec
XCFAS	0006	S	S	SYSTEM	1	0.6	0.0	0.0	0.6	0.0	0.0	0.6	0.6	0.4	0.2	0.0	0.0	366
RMFGAT	0064	S	SO	SYSSTC	1	0.4	0.0	0.0	0.4	0.0	0.0	0.4	0.4	0.4	0.0	0.0	0.0	251
MASTER	0001	S	S	SYSTEM	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.1	0.0	0.0	63
GRS	0007	S	S	SYSTEM	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	48
SMSVSAM	0009	S	S	SYSTEM	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	77
JES2	0051	S	S	SYSSTC	1	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	48
BHBE$XML	0055	B	BO	BTCHDEF	2	0.1	0.0	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	71

z/OS and XML Parsing

▶ z/OS offers two Facilities for low cost and efficient XML Parsing:
  ⇨ z/OS XML Toolkit
  ⇨ z/OS XML System Services

For further details please see the appendix: **z/OS XML Facilities**

XML Parsing with XSL Stylesheets
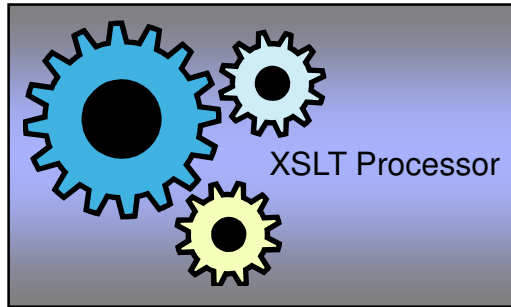
```
<?xml version="1.0">
<ddsml>
  <report>
    <metric id="CPC">
    </metric>
    <caption>
      <var>
        <name>CPCHPNAM</name>
        <value>SYS3</value>
      </var>
      <var>
        <name>CPCHMOD</name>
        <value>2097</value>
      </var>
    </caption>
  </report>
</ddsml>
```

XSLT Processor

Metric;CPC
CPCHPNAM;SYS3
CPCHMOD;2097

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
    <xsl:output omit-xml-declaration="yes" indent="no" method="text"/>
    <xsl:param select="string(';')" name="sep"/>
  - <xsl:template match="/">
        <xsl:apply-templates select="ddsml/report"/>
    </xsl:template>
  - <xsl:template match="report">
        <xsl:apply-templates select="caption/var"/>
        <xsl:text> </xsl:text>
        <xsl:apply-templates select="column-headers/col"/>
        <xsl:apply-templates select="row"/>
        <xsl:text> </xsl:text>
    </xsl:template>
  - <xsl:template match="var">
        <xsl:value-of select="name"/>
        <xsl:value-of select="$sep"/>
        <xsl:value-of select="value"/>
        <xsl:text> </xsl:text>
    </xsl:template>
  - <xsl:template match="row">
        <xsl:apply-templates select="col"/>
    </xsl:template>
  - <xsl:template match="col">
        <xsl:value-of select="."/>
      - <xsl:if test="position()!=last()">
            <xsl:value-of select="$sep"/>
        </xsl:if>
      - <xsl:if test="position()=last()">
            <xsl:text> </xsl:text>
        </xsl:if>
    </xsl:template>
</xsl:stylesheet>
```

Objectives

▶ Generate RMF Monitor III Reports automatically

▶ Provide Sysplex wide Reporting Scope

▶ Store the Reports persistently for each Mintime

▶ Create an Archive for selected Reports

▶ Process individual Reports and apply intelligent Analysis

⇨ Convert the Report XML Document to alternate Formats (CSV, JSON)

⇨ Parse the Report XML Document and extract Key Metrics

▶ Provide a State of the Art Reporting GUI

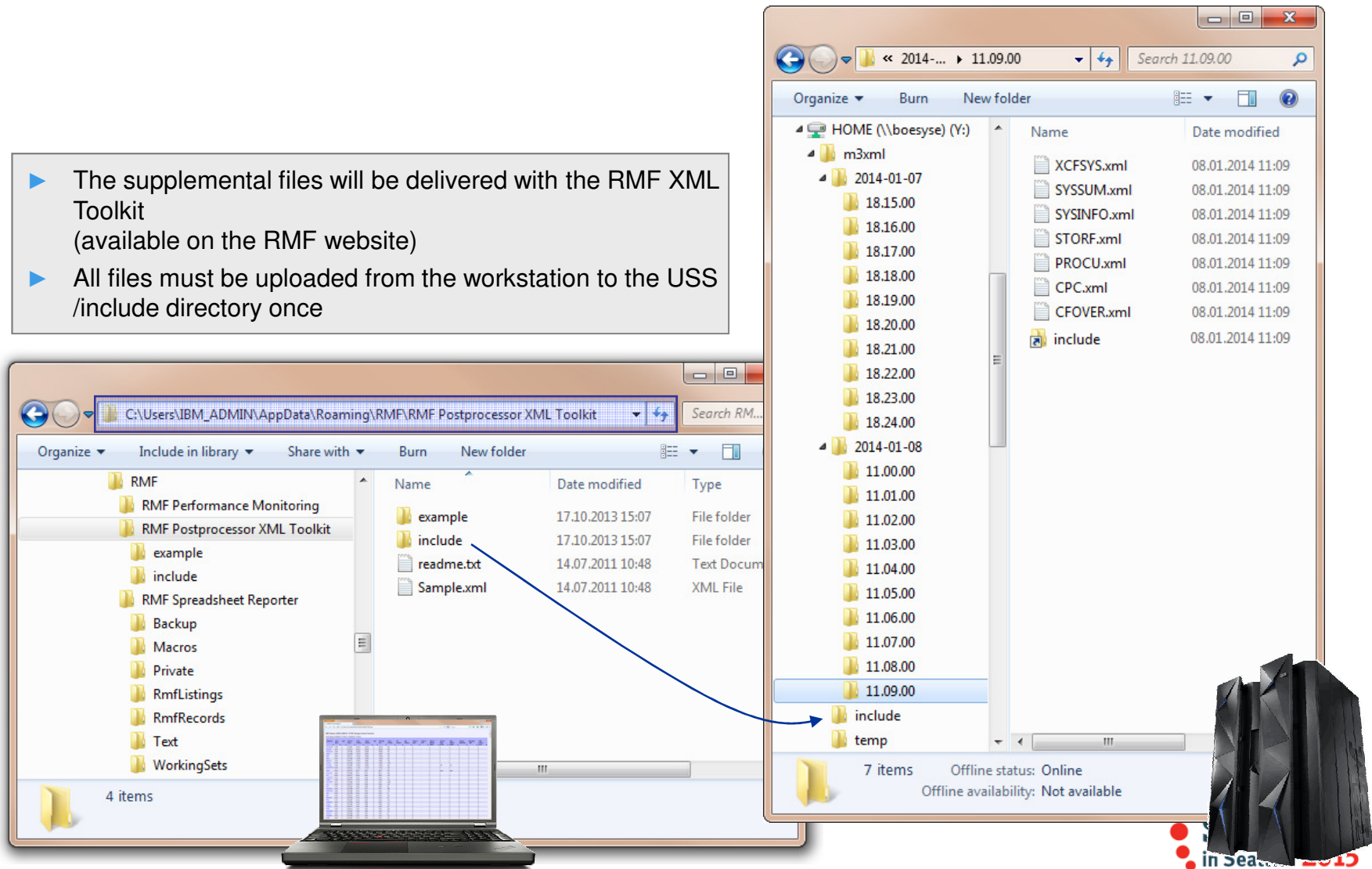▶ Avoid cumbersome Downloads to the Workstation

Mining Gold – The Workstation GUI (1)
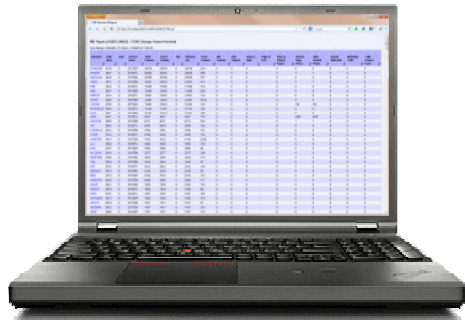

```
file:///D:/m3xml/2014-01-08/11.09.00/STORF.xml - Original Source

File   Edit   Format

 1  <?xml version="1.0" encoding="UTF-8"?>
 2  <?xml-stylesheet type="text/xsl" href="include/ddsml-m3.xsl"?>
 3  <ddsml>
 4  <report>
 5  <metric id="STORF">
 6  <description>STORF (Storage Frames Overview)</description>
 7  <format>report</format>
 8  <numcols>19</numcols>
 9  </metric>
10  <resource>
11  <reslabel>,S4,MVS_IMAGE</reslabel>
12  <restype>MVS_IMAGE</restype>
13  <reslabelurl>%2CS4%2CMVS_IMAGE</reslabelurl>
14  </resource>
15  <time-data>
16  <local-start>20140108110800</local-start>
17  <local-end>20140108110900</local-end>
18  <utc-start>20140108100800</utc-start>
19  <utc-end>20140108100900</utc-end>
20  <local-prev>20140108110730</local-prev>
21  <local-next>20140108110930</local-next>
22  <display-start locale="en-us">01/08/2014 11:08:00</display-start>
23  <display-end locale="en-us">01/08/2014 11:09:00</display-end>
24  <gatherer-interval unit="seconds">60</gatherer-interval>
25  <data-range unit="seconds">60</data-range>
26  </time-data>
27  <row refno="1"><col>DFSKERN</col> <col>0168</col> <col>S</col> <col>STCDEF</col>
```

Additional files are needed to format the XML document :

► XSL   main XSL stylesheet with formatting directives. It is used by the browsers
          XSLT engine to create the HTML document from the XML input
► XSD   XML schema definition
► CSS   cascading stylesheet with additional settings (e.g. fonts, colors)
► JS   java script file with specific processing logic
► GIF   graphical elements like buttons, arrows etc.

Mining Gold – The Workstation GUI (2)

- ▶ All supplemental files are located in the /include directory
- ▶ The m3xml.sh Shell Script creates a symbolic link for each new directory
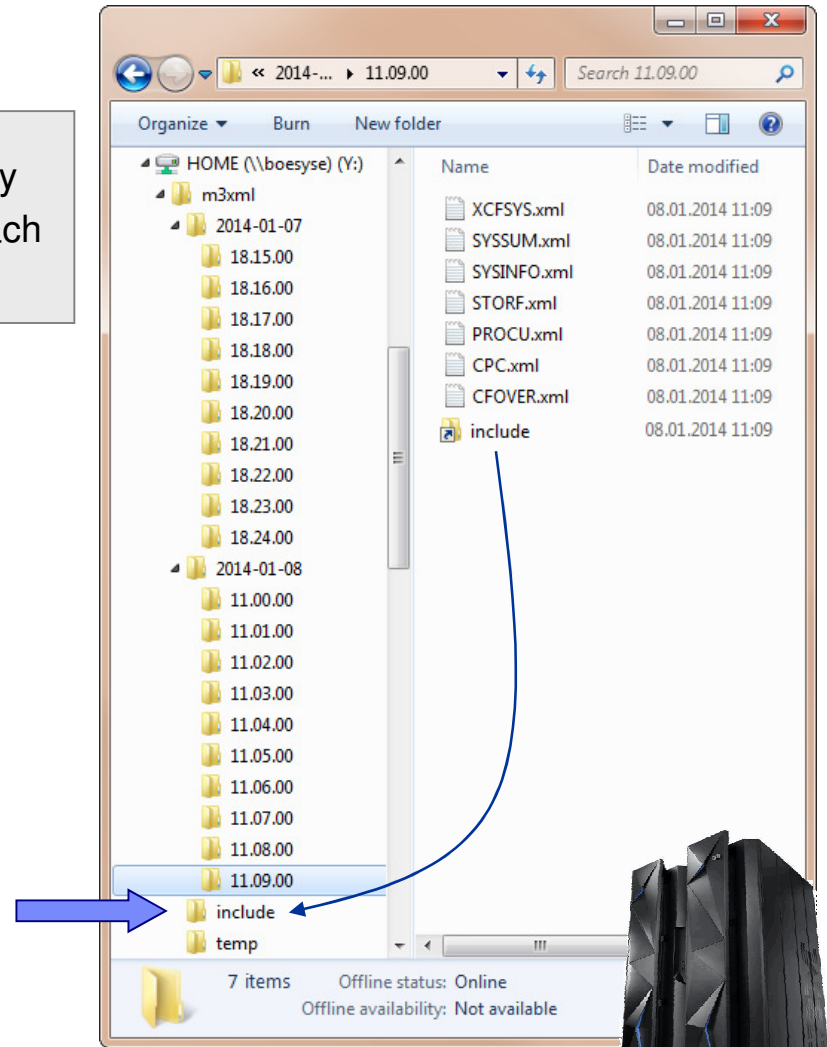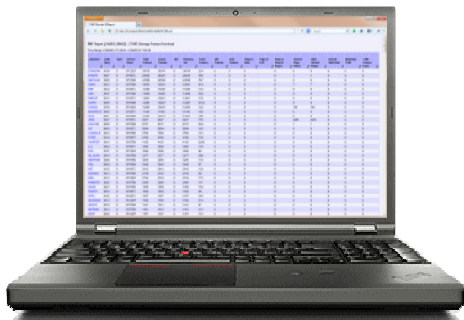
Mining Gold – The Workstation GUI (3)

► The supplemental files will be delivered with the RMF XML Toolkit
(available on the RMF website)

► All files must be uploaded from the workstation to the USS /include directory once

Mining Gold – Reporting with M III VSAM Data Sets (1)

```
//STEP1
//   PARM=(
//GPMINI
//GPMHTC
//GPMPPJCL
//SYSPRINT
//SYSOUT
//RMFDS00
//RMFDS01
//RMFDS02
//RMFDS03
//
```

```
//M3XML     PROC REPORT=,
//              DATE=20140701150000,
//              RANGE=900,
//              UID=0,
//              PWD=0,
//              APPL=0,
//              HOST=0,
//              PORT=0
//*
//GPMM3     EXEC PGM=GPMXMLM3,
//   PARM=('&REPORT &DATE &RANGE &UID &PWD &APPL &HOST &PORT')
//*
//X3RPTS    DD   PATH='/u/rmf/m3xml/temp/&REPORT..xml',
//              PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//              PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//X3XSRPTS DD   PATH= '/u/rmf/m3xml/temp/&REPORT..xml' ,
//              PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//              PATHMODE=(SIRUSR,SIWUSR,SIRGRP),FILEDATA=TEXT
//SYSPRINT DD   SYSOUT=*
//SYSOUT    DD   SYSOUT=*
//          PEND
```

Monitor III VSAM Datasets can be supplied
to the DDS with the same DD Names as
for a Monitor III ISPF Report Session

Mining Gold – Reporting with M III VSAM Data Sets (2)

Mining Gold – Realtime Parsing (1)

Mining Gold – Realtime Parsing (2)


```c
int cfact   (char *repname, char *resource, char *xml) {
    int rc = 1;
    PARSEXML(rc, repname, resource, xml);
}
int cfover   (char *repname, char *resource, char *xml) {
    int rc = 1;
    PARSEXML(rc, repname, resource, xml);
}
int cfsys   (char *repname, char *resource, char *xml) {
    int rc = 1;
    PARSEXML(rc, repname, resource, xml);
}
```

Module GPMXMLM3 calls exit module GPMXMLX3

Mining Gold – Realtime Parsing (3)

PGM=GPMXMLM3

> Module GPMXMLM3 is shipped with RMF V2R1

► Obtains the Monitor III XML report from RMF DDS (based on parameters)

► Tries to load exit module GPMXMLX3

► Searches for a function name in GPMXMLX3 equal to the report type

► If entry point could be found calls the function

► The parameters are:
  ⇨ Report name
  ⇨ Resource qualifier (eg. *,,SYSPLEX* or *,SYSA,MVS_IMAGE*)
  ⇨ XML Document

► Writes the XML report to the USS file system (based on the return code of the GPMXMLX3 function)

PGM=GPMXMLX3

> Module GPMXMLX3 is shipped with RMF V2R1. A source code example is planned for RMF V2R2.

► Provides one function skeleton per report type

► Can parse the XML document and perform an individual action (CSV conversion, WTO...)

► Sets the return code in order to indicate whether the report should be written (rc=1) or not (rc=0)

Summary

The Monitor III XML Batch Facility can:

► Generate RMF Monitor III Reports automatically

► Provide Sysplex wide Reporting Scope

► Store the Reports persistently for each Mintime

► Create an Archive for selected Reports

► Process individual Reports and apply intelligent Analysis

⇨ Convert the Report XML Document to alternate Formats (CSV, JSON)

⇨ Parse the Report XML Document and extract Key Metrics

► Provide a State of the Art Reporting GUI

► Avoid cumbersome Downloads to the Workstation

Appendix – z/OS XML Facilities

- ❑ XML Processing in the z/OS environment
  - ❍ z/OS XML Toolkit
  - ❍ z/OS XML System Services
- ❑ XML Parsing with z/OS XML System Services
  - ❍ Callable Services Overview
  - ❍ Structure of Parsed Documents
  - ❍ C++ Mappings and Coding Example

z/OS XML Toolkit

- ❑ Optional Priced Feature (FMID HXML 1A0)
- ❑ Two Components
  - ○ XML Parser
  - ○ XSLT Processor
- ❑ Advanced Parsing Functions: Full implementation of SAX and DOM Parsers
- ❑ Stubs and Modules in the USS File System and in z/OS Library Format
- ❑ USS File System
  - ○ /usr/lpp/ixm/IBM/xml4c-5_7   XML Parser
  - ○ /usr/lpp/ixm/IBM/xslt4c-1_11   XSLT Processor

📁 bin		15.10.2008	rwxr-xr-x	RMF
📁 doc		14.10.2008	rwxr-xr-x	RMF
📁 include		15.10.2008	rwxr-xr-x	RMF
📁 lib		15.10.2008	rwxr-xr-x	RMF
📁 samples		15.10.2008	rwxr-xr-x	RMF
📁 zsamples		15.10.2008	rwxr-xr-x	RMF
📄 KEYS	9.114	14.10.2008	rw-r--r--	RMF
📄 LICENSE	11.601	14.10.2008	rw-r--r--	RMF
📄 NOTICE	560	14.10.2008	rw-r--r--	RMF
📄 license.html	644	14.10.2008	rw-r--r--	RMF
📄 Readme.html	1.017	14.10.2008	rw-r--r--	RMF

- ❑ z/OS Library Format
  - ○ Stubs   SYS1.SIXMEXP
  - ○ Run-Time Library   SYS1.SIXMLOD1

z/OS XML System Services

- Integrated z/OS Component
- Provides set of services
  - gxlpQuery
  - gxlpInit
  - gxlpParse
  - gxlpTerminate
- Header Files   SYS1.SIEAHDRV.H
  - GXLHXML   Function prototypes
  - GXLHQXD   Query service return data mapping
  - GXLHXEH   Output record data mappings
  - GXLHXEC   Constants
- Link Library: SYS1.SIEASID(GXLXXML1/GXLXXML4)
- Run-Time Library: SYS1.SIEALNKE(GXLCXML1/GXLCXML4)

z/OS XML System Services

gxlpQuery — query an XML document

Obtains the XML characteristics of a document. The XML characteristics are either the default values, the values contained in an XML declaration or a combination of both.

```
int gxlpQuery (void * work_area,
          long work_area_length,
          void * input_buffer,
          long input_buffer_length,
          GXLHQXD ** return_data,
          int * rc_p,
          int * rsn_p);
```

Return Data is mapped by GXLHQXD.H

```
typedef struct _GXLHQXD
{
    int QXD_Version;
    unsigned int QXD_XML_Autodet_value;
    unsigned int QXD_XML_Autodet_CCSID;
    unsigned short QXD_XML_Version;
    unsigned short QXD_XML_Release;
    unsigned int QXD_XML_Specified_CCSID;
    unsigned char QXD_XML_Flag1;
    unsigned char QXD_XML_Flag2;
    unsigned short Rsvd_18;
    unsigned int QXD_XML_Decl_Len;
} GXLHQXD;
```

XML Version & Release

Encoding Information

```
<?xml version="1.0" encoding="UTF-8"?>
<directory>
 <user id="007">
  <name>
   <first>James</first>
   <last>Bond</last>
  </name>
  <profession>problem solver</profession>
 </user>
</directory>
```
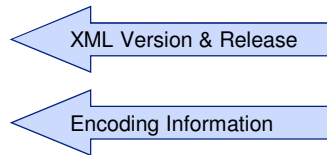
z/OS XML System Services

gxlpInit — initialize the XML Parser

Initializes the PIMA and records the addresses of the caller's system service routines (if any).
The PIMA storage is divided into the areas that will be used by the XML parser to process the input buffer and produce the parsed data stream.

```
int gxlpInit (void * PIMA,              <-- Parse Instance Memory Area
              long PIMA_LEN,
              int ccsid,                <-- Coded Character Set Identifier
              int feature_flags,
              GXLHXSV sys_svc_vector,   <-- Exit  Routines and Parameters
              void * sys_svc_parm,
              int * rc_p,
              int * rsn_p);
```

z/OS XML System Services

gxlpParse — parse a buffer of XML text

Parses a buffer of XML text and places the result in an output buffer.

```
int gxlpParse(void * PIMA,
    int * option_flags,
    void ** input_buffer_addr,
    long * input_buffer_bytes_left,
    void ** output_buffer_addr,
    long * output_buffer_bytes_left,
    int * rc_p,
    int * rsn_p);
```

Parse Instance Memory Area

XML Input Document

Parsed Result Area
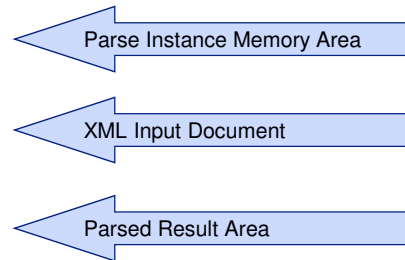
gxlpTerminate — terminate a parse instance

Releases all resources obtained (including storage) by the XML parser and resets the PIMA so that it can be re-initialized or freed.

```
int gxlpTerminate (void * PIMA,
    int * rc,
    int * rsn);
```

Parse Instance Memory Area
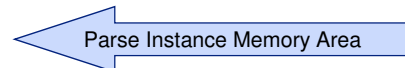
z/OS XML System Services – Parser Example

SHARE
Educate · Network · Influence

```
<?xml version="1.0" encoding="UTF-8"?>
<directory>
 <user id="007">
  <name>
   <first>James</first>
   <last>Bond</last>
  </name>
  <profession>problem solver</profession>
 </user>
</directory>
```
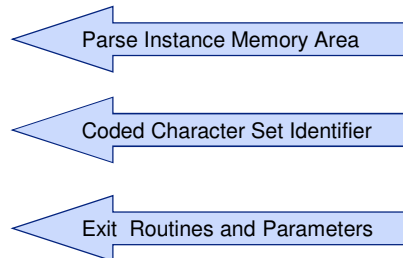
gxlpParse()

```
0000000:    F00F00000000020000000000000000000    *0..............*
+00000010:  00000000000001680000000000000000000  *...............*
+00000020:  F01F00000000001C00000003F14BF000     *0...........1.0.*
+00000030:  000005E4E3C660F800000000F02F0000     *...UTF-8....0...*
+00000040:  0000001D00000009848999858 3A39699    *........director*
+00000050:  A8000000000000000000F02F0000000000   *y........0......*
+00000060:  1800000004A4A2859900000000000000     *.....user.......*
+00000070:  00F04F0000000000160000002898400      *.0|..........id.*
+00000080:  00000000000000F05F40000000000F00     *.......0^ ......*
+00000090:  000003F0F0F7F02F0000000000180000     *...0070.........*
+000000A0:  00049581948500000000000000000F02F    *..name.........0.*
+000000B0:  00000000001900000005868999A2A300     *..........first.*
+000000C0:  0000000000000F07F40000000001100      *.......0" ......*
+000000D0:  000005D1819485A2F03F000000000008     *...James0.......*
+000000E0:  F02F0000000000180000000049381A2A3    *0...........last*
+000000F0:  0000000000000000F07F400000000010     *........0" .....*
+00000100:  00000004C2969584F03F000000000008     *....Bond0.......*
+00000110:  F03F0000000000008F02F00000000001E    *0.......0.......*
+00000120:  0000000A9799968685A2A28996950000     *....profession..*
+00000130:  000000000000F07F40000000001A0000     *......0" .......*
+00000140:  000E9799968293859440A29693A58599    *..problem solver*
+00000150:  F03F0000000000008F03F000000000008    *0.......0.......*
+00000160:  F03F0000000000008                    *0.......       *
```

```
+00000000:    F00F000000000200000000000000000    *0..............*
+00000010:    00000000000001680000000000000000    *................*
+00000020:    F01F00000000001C00000003F14BF000    *0...........1.0.*
+00000030:    000005E4E3C660F800000000F02F0000    *...UTF-8....0...*
+00000040:    0000001D000000098489998583A39699    *.........director*
+00000050:    A8000000000000000F02F0000000000    *y........0......*
+00000060:    1800000004A4A28599000000000000000    *.....user.......*
+00000070:    00F04F000000000160000000002898400    *.0|..........id.*
+00000080:    0000000000000F05F40000000000F00    *.......0^ ......*
+00000090:    000003F0F0F7F02F0000000000180000    *...0070.........*
+000000A0:    0004958194850000000000000000F02F    *..name........0.*
+000000B0:    000000000019000000005868999A2A300    *..........first.*
+000000C0:    0000000000000F07F40000000001100    *.......0" ......*
+000000D0:    000005D1819485A2F03F000000000008    *...James0.......*
+000000E0:    F02F0000000001800000049381A2A3    *0...........last*
+000000F0:    0000000000000000F07F40000000010    *........0" .....*
+00000100:    00000004C2969584F03F000000000008    *....Bond0.......*
+00000110:    F03F000000000008F02F00000000001E    *0.......0.......*
+00000120:    0000000A9799968685A2A28996950000    *....profession..*
+00000130:    000000000000F07F40000000001A0000    *......0" .......*
+00000140:    000E9799968293859440A29693A58599    *..problem solver*
+00000150:    F03F000000000008F03F000000000008    *0.......0.......*
+00000160:    F03F000000000008                   *0.......        *
```

GXLHXEC_TOK_BUFFER_INFO   0xF00F

GXLHXEC_TOK_XML_DECL   0xF01F

GXLHXEC_TOK_START_ELEM   0xF02F

GXLHXEC_TOK_START_ELEM   0xF02F

GXLHXEC_TOK_ATTR_NAME   0xF04F

GXLHXEC_TOK_ATTR_VALUE   0xF05F

GXLHXEC_TOK_START_ELEM   0xF02F

GXLHXEC_TOK_CHAR_DATA   0xF07F

GXLHXEC_TOK_END_ELEM   0xF03F
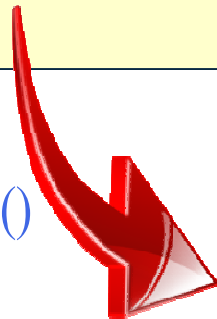
```xml
<?xml version="1.0" encoding="UTF-8"?>
<directory>
 <user id="007">
  <name>
   <first>James</first>
   <last>Bond</last>
  </name>
  <profession>problem solver</profession>
 </user>
</directory>
```

z/OS XML System Services – Parser

```
+00000000:    F00F0000000002C0000000000000000    *0..............*
+00000010:    000000000000016800000000000000    *...............*
+00000020:    F01F00000000001C0000003F14BF000    *0...........1.0.*
+00000030:    000005E4E3C660F8000000000F02F0000   *...UTF-8....0...*
+00000040:    0000001D0000000984899598583A39699   *........director*
+00000050:    A80000000000000000F02F0000000000    *y........0......*
+00000060:    1800000004A4A285990000000000000    *.....user.......*
+00000070:    00F04F000000000016000000002898400   *.0|..........id.*
+00000080:    0000000000000F05F40000000000F00    *.......0^ ......*
+00000090:    000003F0F0F7F02F0000000000180000    *...0070.........*
+000000A0:    0004958194850000000000000000F02F    *..name........0.*
+000000B0:    00000000001900000005868999A2A300    *..........first.*
+000000C0:    00000000000000F07F40000000001100    *.......0" ......*
+000000D0:    000005D1819485A2F03F000000000008    *...James0.......*
+000000E0:    F02F0000000001800000049381A2A3    *0...........last*
+000000F0:    00000000000000F07F40000000010    *........0" .....*
+00000100:    00000004C2969584F03F000000000008    *....Bond0.......*
+00000110:    F03F000000000008F02F00000000001E    *0.......0.......*
+00000120:    0000000A9799968685A2A28996950000    *....profession..*
+00000130:    000000000000F07F40000000001A0000    *......0" .......*
+00000140:    000E97999682938594440A29693A58599    *..problem solver*
+00000150:    F03F000000000008F03F000000000008    *0.......0.......*
+00000160:    F03F000000000008    *0.......      *
```

Total Length of Document

Length of Element

```
<?xml version="1.0" encoding="UTF-8"?>
<directory>
 <user id="007">
  <name>
   <first>James</first>
   <last>Bond</last>
  </name>
  <profession>problem solver</profession>
 </user>
</directory>
```

z/OS XML System Services – Output Area

Return Data is mapped by GXLHQXD.H

```
typedef struct _GXLHXEH_RECORD
{
    unsigned short XEH_TokType;
    unsigned char XEH_Flags;
    unsigned char XEH_Reserved;
    int XEH_RecLen;
    char XEH_Values;
} GXLHXEH_RECORD;
```

```
+00000000:    F00F000000002000000000000000000    *0..............*
+00000010:    0000000000000168000000000000000    *...............*
+00000020:    F01F00000000001C00000003F14BF000   *0...........1.0.*
```

```
typedef struct _GXLHXEH_BUFINFODATA
{
    unsigned int XEH_DSOpts;
    unsigned short XEH_PrsStat;
    unsigned short XEH_BufRsv;
    unsigned long XEH_BufLenUsed;
    unsigned long XEH_ErrOffset;
} GXLHXEH_BUFINFODATA;
```

```
+00000000:    F00F000000002000000000000000000    *0..............*
+00000010:    0000000000000168000000000000000    *...............*
+00000020:    F01F00000000001C00000003F14BF000   *0...........1.0.*
```

```
typedef struct _GXLHXEH_VALUE
{
    int XEH_ValLen;
    char XEH_ValText;
} GXLHXEH_VALUE;
```

```
+000000E0:    F02F000000000018000000049381A2A3   *0...........last*
+000000F0:    0000000000000000F07F400000000010   *........0" .....*
+00000100:    00000004C2969584F03F000000000008   *....Bond0.......*
+00000110:
```

z/OS XML System Services – Parser

```
//----------------------------------------------------------------
//
//   C++ Programming Example:
//   - loops through XML Output Buffer
//   - prints all tag names and character data
//
//----------------------------------------------------------------

char text[256];
int  textLen;
GXLHXEH_RECORD *pXEHRecord = output_buffer_addr;
GXLHXEH_RECORD *pXEHEnd = pXEHRecord + (GXLHXEH_RECORD *)pXEHRecord->XEH_BufLenUsed;
GXLHXEH_VALUE  *pXEHValue;
while (pXEHRecord < pXEHEnd) {
    pXEHValue = (GXLHXEH_VALUE *)&pXEHRecord->XEH_Values;
    textLen = pXMLValue->XEH_ValLen;

    switch (pXEHRecord->XEH_TokType) {
        case GXLHXEC_TOK_START_ELEM:
        case GXLHXEC_TOK_CHAR_DATA:
            memcpy(text, (char *)&pXEHValue->XEH_ValText, textLen);
            *(text + pXMLValue->XEH_ValLen) = '\0';
            printf("String found: %s\n, text);
            break;
        case GXLHXEC_TOK_END_ELEM:
        break;
        default:
        break;
    }
    pXMLRecord =
    (GXLHXEH_RECORD *)((char *)pXEHRecord + pXEHRecord->XEH_RecLen);
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<directory>
 <user id="007">
  <name>
   <first>James</first>
   <last>Bond</last>
  </name>
  <profession>problem solver</profession>
 </user>
</directory>
```

Summary

The Monitor III XML Batch Facility can:

▶ Generate RMF Monitor III Reports automatically

▶ Provide Sysplex wide Reporting Scope

▶ Store the Reports persistently for each Mintime

▶ Create an Archive for selected Reports

▶ Process individual Reports and apply intelligent Analysis

  ⇨ Convert the Report XML Document to alternate Formats (CSV, JSON)

  ⇨ Parse the Report XML Document and extract Key Metrics

▶ Provide a State of the Art Reporting GUI

▶ Avoid cumbersome Downloads
   to the Workstation

```
<?xml version="1.0" encoding="UTF-8"?>
<directory>
 <user id="007">
  <name>
   <first>James</first>
   <last>Bond</last>
  </name>
  <profession>problem solver</profession>
 </user>
</directory>
```

Information and Tools

RMF website: www.ibm.com/systems/z/os/zos/features/rmf/

- Product information, newsletters, presentations, ...
- Downloads
  - ►Spreadsheet Reporter
  - ►RMF PM Java Edition
  - ►Postprocessor XML Toolkit

RMF email address: rmf@de.ibm.com

Documentation and news:

- ⇨RMF Report Analysis, SC34-2665
- ⇨RMF User's Guide, SC34-2664
- ⇨PDF files can be downloaded from:
  www.ibm.com/systems/z/os/zos/bkserv