

# **SYSchange: The Next Generation of Disaster Recovery Software**

*By Michael Madani*

During a disaster recovery (DR), real life often offers surprises! Frequently, an important JCL member will be missing, an old version of software will be running, or an entire library will be missing because the restore job failed the last time it was run. Without a good systems software solution in place, such incidents often go unnoticed at the expense of the organization.

No matter how many times a DR plan has been tested, there are always lingering questions, such as, "Do we have all the software that we need to run production?" and "Will we be able to run our business critical applications successfully off the DR system, with the same results as running off production?" **SYSchange**, the next generation of systems software change management, is designed to address such concerns in z/OS environments. **SYSchange** not only verifies software integrity between your DR and production system within minutes, but also maintains your DR software at the same level as your production software. Using the newest technologies delivered by **SYSchange**, these tasks can be performed regularly with minimal overhead and in an automated fashion. Additionally, **SYSchange** provides an innovative Started Task (STC) technology that transparently backs up changed library components every time a change is introduced, regardless of how the change was made. This automatic member-level backup capability, inherent in **SYSchange**, is critical to recovery.

## **How does SYSchange verify your DR software integrity?**

Using the **SYSchange** 'tokenization' technology, users can TOKENIZE any nonVSAM data set to identify changes between environments, and to detect changes made to an environment over time. This sophisticated process is highly efficient with no impact to the system. Tokenization quickly establishes a 4-byte value representing the content of each member of a partitioned data set (source or load). A single token value is generated to represent an entire physical sequential or a direct access file. These token values are internally stored in the **SYSchange** Control File for future change detection. At a later point in time, when any two software environments need to be compared, the environment of interest is re-tokenized and compared against the stored tokens. Any changes since the original tokens were generated are immediately identified, reported, and optionally packaged for distribution. In **SYSchange**, tokenization forms the basis for comparing local or remote software environments. The following steps provide a practical example of such a scenario.

### **Step 1: Tokenize your production software**

The following commands tokenize a sample production set of libraries:

```
TOKENIZE=(DSN=PROD.PAYROLL.JCL)
TOKENIZE=(DSN=PROD.PAYROLL.LINKLIB)
TOKENIZE=(DSN=PROD.PAYROLL.CLIST)
....
```

**SYSchange** also allows another way to tokenize a software environment, using the PATTERN command as shown:

```
PATTERN=PROD.PAYROLL.*
PATTERN=PROD.ONLINE.*
PATTERN=PROD.*.LINKLIB
*
TOKENIZE
```

### ***Step 2: Restore the SYSchange Control File and Data File at the DR site***

We assume that you have a running DR site (a hot site), or you have just restored the DR software from full volume dumps to DASD, and now before you start your production you intend to verify the integrity of your DR software with production. To establish a representation of your production environment at your DR site, you need to restore the two **SYSchange** required VSAM files: the Control File and the Data File. These files contain the tokens of your production software generated in Step 1.

### ***Step 3: Use the MODS parameter to verify the integrity of your DR software***

The following commands must be run at your DR site. **SYSchange** tokenizes the specified data sets and compares the tokens with the ones stored in the **SYSchange** Control File (representing your production software).

```
OPTION=MODS
*
TOKENIZE=(DSN=PROD.PAYROLL.JCL,MODS)
TOKENIZE=(DSN=PROD.PAYROLL.LINKLIB,MODS)
TOKENIZE=(DSN=PROD.PAYROLL.CLIST,MODS)
....
```

Note the additional MODS parameter specified in the TOKENIZE command. When the MODS parameter is specified, **SYSchange** uses only the newly generated tokens for comparison but does not store them in the Control File (as in the case when TOKENIZE is used without the MODS parameter). The OPTION=MODS command reports the MISMATCH DR library members (members existing in one library and not the other). MODS also reports DIFF members (those members matching in name but not contents). DR administrators should pay close attention to the report and resolve the differences to ensure a successful disaster recovery. If OPTION=MISMATCH is specified, **SYSchange** reports only mismatch members, and other exceptions are not reported. Similarly, OPTION=DIFF reports only those members which have different content. As before, the PATTERN command may be used to specify a larger environment as shown below:

```
OPTIONS=MODS
*
PATTERN=PROD.PAYROLL.*
PATTERN=PROD.ONLINE.*
```

```
PATTERN=PROD.*.LINKLIB  
  
*  
  
SHOWMODS
```

## Keeping your DR software and production software at the same level

As mentioned earlier, **SYSchange** not only enables your data center to regularly and cost-effectively compare your DR software with your production software, but also allows you to rapidly identify and package only changed components, thereby dramatically increasing efficiency while reducing the time and costs associated with such tasks. In **SYSchange**, the Promotion Package is used as the vehicle for distribution of all your software changes to your DR site. Executable load modules, source members, JCL, PROCS, CLISTS, PANELS, etc., are all packaged together into this single Promotion Package. Once a package is created, you can use a transfer program of your choice such as the MVS TSO XMIT, NDM, FTP, or any other data mover software to transfer the file. Once a Promotion Package is received at the DR site, it can be promoted to update the target libraries. Alternatively, a DR administrator can keep such packages and use them when the need arises. In this fashion, daily changes are quickly put into the **SYSchange** Promotion Package and transferred to your DR site. Identifying and packaging changes is a two-part process within a single job as shown:

```
//JOB  
//*  
/* _____  
/* identify changed data sets and pass the DSnames to the next step  
/* -----  
//IDENTIFY EXEC PGM=SYCMAN,REGION=4096K  
//OUTFILE DD DISP=SHR,DSN=somenode.SYC.CHGDSNS <== input for 2nd step  
//SYSPRINT DD SYSOUT=*  
//SYCIN DD *
```

```
OPTION=PACKAGE
```

```
*
```

```
PATTERN=PROD.PAYROLL.*
```

```
PATTERN=PROD.ONLINE.*
```

```
PATTERN=PROD.*.LINKLIB
```

```
...
```

```
IDENTIFY
```

```
/*  
/* _____  
/* Package the MODS  
/* -----  
//PACKAGE EXEC PGM=SYCMAN,REGION=4096K  
//SYSPRINT DD SYSOUT=*  
//SYCIN DD DISP=SHR,DSN=somenode.SYC.PKGCMSD <== commands for packaging  
// DD DISP=SHR,DSN=somenode.SYC.CHGDSNS <== DSNs from 1st step
```

Assuming that your production libraries have already been tokenized, the commands in the first step identify the data sets that have some changed components. Typically, from hundreds or thousands of production libraries, only a small subset changes daily. Regardless of how a change is introduced, standard (using IEBCOPY, IEBGENER, LINKEDIT, ISPF, etc.) or non-standard (using

full track I/Os or super zaps), **SYSChange** identifies the changed data sets using its tokenization technology. The changed data sets are then passed to the next step for member-level change identification and packaging.

### Update your DR software daily using the Promotion Package

**SYSChange** allows you to view the content of a Promotion Package using the SCAN command. All offloaded components in the package are reported in a similar fashion to ISPF 3.1 with their ISPF status included in the report. Once the Promotion Package is transferred to your DR site using the PROMOTE command, you can update your DR software environment as shown below:

```
PROMOTE=( DD=DD1 ,  
          FEEDBACKCOMMENT='DR SITE SOFTWARE UPDATED SUCCESSFULLY' ,  
          SELECT=( DSN=* , MEM=* ) )
```

### Conclusion

**SYSChange** provides organizations with next-generation technologies for verifying and maintaining system software integrity with absolute certainty. The versatility and simplicity of **SYSChange** ensures that the job gets done with minimal effort and minimal impact. **SYSChange** developers are cognizant that since DR tasks must be performed on a daily basis, organizations require greater levels of automation and transparency in their next-generation solutions. **SYSChange** achieves this objective without impinging on your existing solutions and methodologies. Furthermore, **SYSChange** protection mechanisms have been designed to provide the greatest immunity against unauthorized change, while ensuring the health and recoverability of your systems.

#### **Michael Madani – Pristine Software**

*Michael Madani is the Founder and Chief Architect at Pristine Software. Pristine has provided corporations around the world with leading-edge software solutions, with a focus on Automating System Software Change Management, since 1989. Michael's career has spanned more than 25 years in mainframe software development.*

Mainstar is a registered trademark of Mainstar Software Corporation. SYSChange is a registered trademark of Pristine Software. IBM is a registered trademark and DFSMSHsm is a trademark of the International Business Machines Corporation.