Verify/alloc/delete of BAS- PROM- or PROD DSNs

- <Built-in Solutions> |
- <XML services in CMN/ZMF> |
- <CMN/ZMF Application Administration>

Summary: Create a set of baskets for a single application (alternative: for all applications). Each basket contains all DSNs defined in application administration as production DSNs, baseline DSNs or promotion DSNs.

The **set of baskets** created are:

- BsktPrfx.Subsys.DSNADMUT.Date.Time.Subsys.PrmSite for each promotion site containing all (at least 1) shadow promotion DSNs and local promotion DSNs of all promotion levels on this site as defined via A.A.7.
- BsktPrfx.Subsys.DSNADMUT.Date.Time.Rmtsys.PrdSite for each production site containing all (at least 1) production DSNs on this site as defined via A.A.P and eventually also (if the site is also used for promotion) containing all remote promotion DSNs on this site as defined via A.A.7.
- BsktPrfx.Subsys.DSNADMUT.Date.Time.Subsys.BASELIB, for each baseline DSN (0 or –X) as defined via A.A.B.

In the **DSNs** mentioned above, these **conventions** were used:

- BskPrf: some high level qualifier for the basket DSNs to be created.
- Subsys: the CMN/ZMF subsystem ID of the DP site (that contains all A.A definitions).
- Date: a qualifier corresponding to the date the baskets were created.
- Time: a qualifier corresponding to the time the baskets were created.
- PrmSite: the name of the promotion site (as defined in A.A.7).
- PrdSite: the name of the production site (as defined in A.A.P).

The **content of each basket** (record structure) depends on the selected processing option:

- The first basket record contains the XML tag “mvslib” and each subsequent basket record consists of only a DSN (for processing option “info” or “delete”).
- The first basket record contains all XML input tags for XML service DSS/SERVICE/ALLOCATE and each subsequent record consists of a DSN followed by the DSN’s actual values (derived from the attributes stored in A.A.2) for each of these input tags (for processing option “allocate”).

The baskets created using processing option “info” or “delete” can be used as input for either of the following (green) XML services:

- DSS/SERVICE/INFO, to verify the existence of each of the DSNs included in the basket.
- DSS/SERVICE/DELETE, to delete each of the DSNs included in the basket.

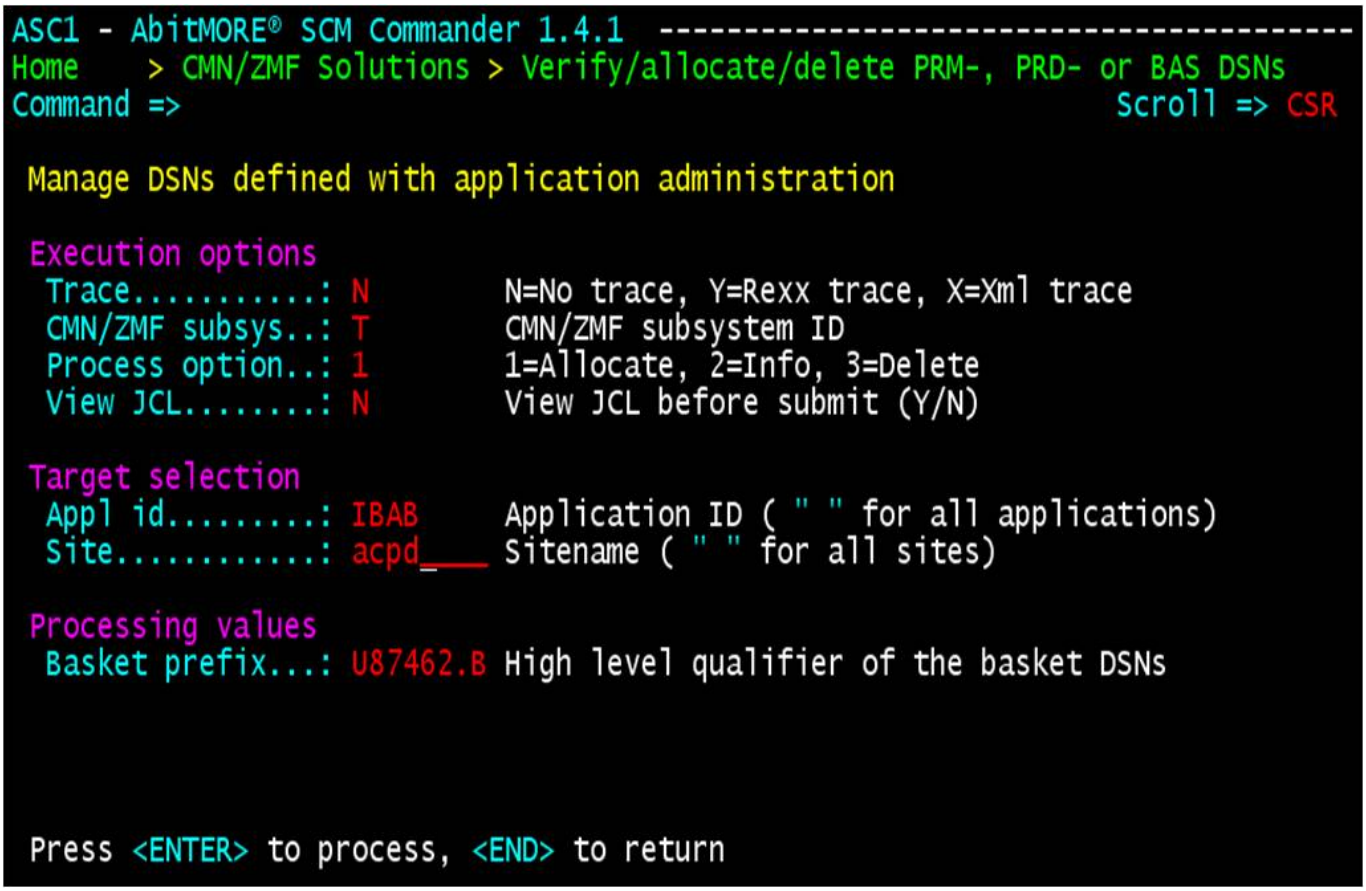
The baskets created using processing option “allocate” can be used as input for the (green) XML service DSS/SERVICE/ALLOCATE, to allocate each of the DSNs included in the basket.

To actually execute any of these XML services, just use ASC’s basket processing features. It may be a good idea also to use “Stop on error = N” during this basket processing.

In case the DSNs to be processed (as contained in the baskets) reside on a remote site, following options exist:

- If TCP/IP is enabled, just use ASC's TCP/IP facilities (available for basket processing also), using the CMN/ZMF subsystem ID that manages the remote site DSNs as the subsystem to execute the XML services.
- If TCP/IP is not enabled just transfer the baskets to the remote site, and then launch ASC on the remote site to process the transferred baskets.

Solution ID: ASCZ0202
Solution Variables:

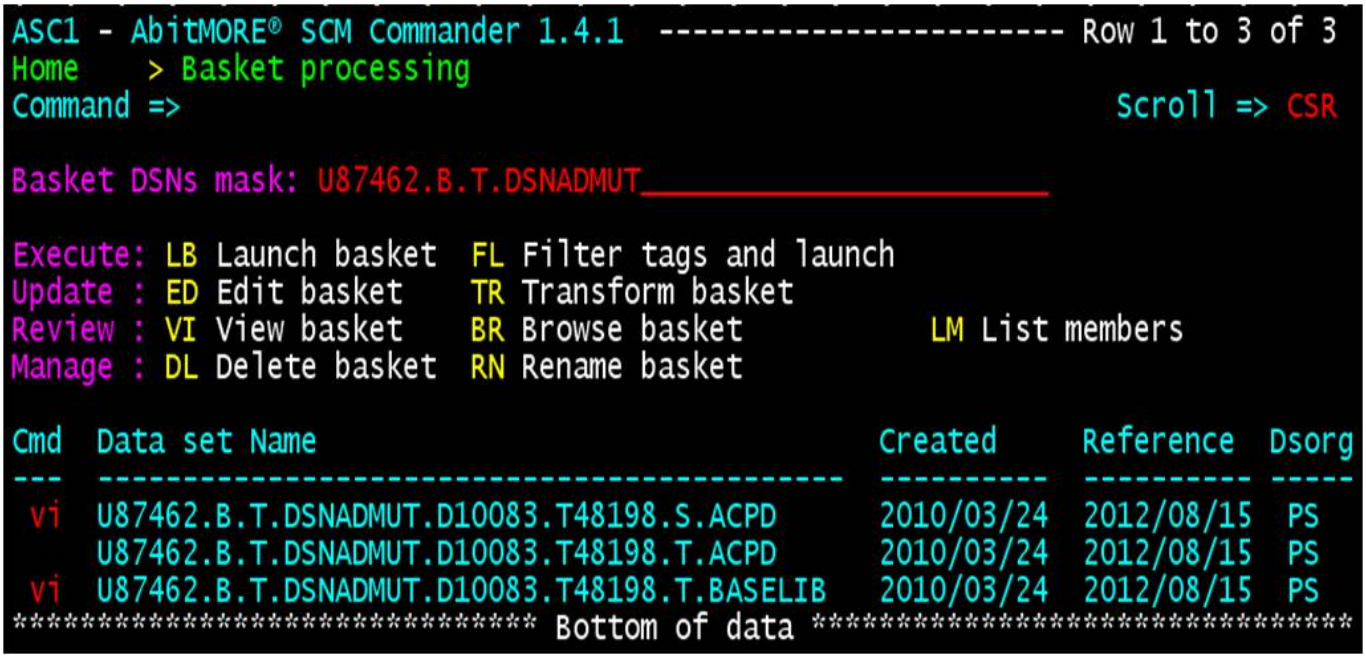


The above screen (via **Appl id** within **Target selection**) shows that this solution can be used for just 1 application, or for all (dozens? hundreds?) applications in 1 shot.

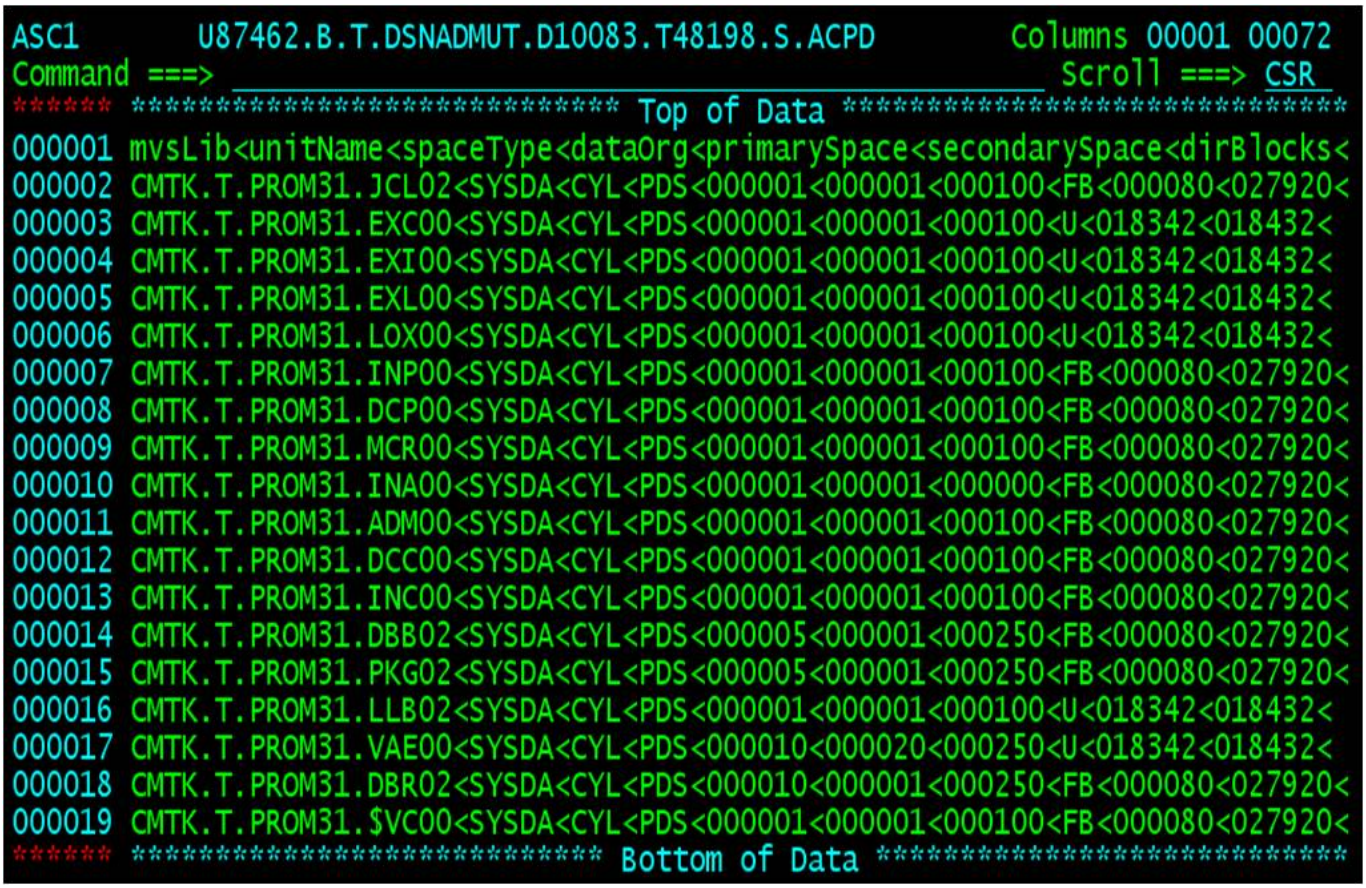
Also note that this solution (via **Site** within **Target selection**) can be used to process "ALL" sites together, or to process just 1 site (e.g. a new promotion or production target added in various applications) and eventually to be repeated for a 2nd site.

To understand how the output of this solution looks like, and how to further process that output, checkout the various steps in the scenario below.

Scenario:
Step 01: List the baskets created by ASCZ0202



Step 02: View the baskets content (1/2)



This is a perfect illustration of how "real" baskets look like (= CSV-kind of files).

The various data included in each record, is what gets created by using a **Process option** (within **Execution options**) with value 1 (Allocate), because all these data are required as input tags for the XML service to actually **allocate** these DSNs (shown in step 04 below).

By using value 1 (Info) or 3 (Delete), each record will only contain the mvslib value, because only a DSN is required as input tag for the XML service to either report about the DSN info (with data similar to ISPF option 3.2) or to actually **delete** these DSNs.

Step 02: View the baskets content (2/2)

```
ASC1       U87462.B.T.DSNADMUT.D10083.T48198.T.BASELIB     Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ******************************** Top of Data ********************************
000001 mvsLib<unitName<spaceType<dataOrg<primarySpace<secondarySpace<dirBlocks<
000002 CMTT.T.BASE0.ADM00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
000003 CMTT.T.BASE1.ADM00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
000004 CMTT.T.BASE0.$SX00.$$$$<SYSDA<CYL<PDS<000001<000001<000100<FB<000133<000
000005 CMTT.T.BASE1.$SX00.$$$$<SYSDA<CYL<PDS<000001<000001<000100<FB<000133<000
000006 CMTT.T.BASE0.EXE00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000007 CMTT.T.BASE1.EXE00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000008 CMTT.T.BASE0.LOD00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000009 CMTT.T.BASE1.LOD00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000010 CMTT.T.BASE0.LOX00.$$$$<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000011 CMTT.T.BASE1.LOX00.$$$$<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000012 CMTT.T.BASE0.EXC00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000013 CMTT.T.BASE1.EXC00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000014 CMTT.T.BASE0.EXI00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000015 CMTT.T.BASE1.EXI00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<U<018342<0184
000016 CMTT.T.BASE0.JCL00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
000017 CMTT.T.BASEN.JCL00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
000018 CMTT.T.BASE0.SRC00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
000019 CMTT.T.BASEN.SRC00.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  108 Line(s) not Displayed
000128 CMTT.T.BASE0.$VC02.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
000129 CMTT.T.BASEN.$VC02.IBAB<SYSDA<CYL<PDS<000001<000001<000100<FB<000080<027
****** ******************************** Bottom of Data ********************************
```

Step 03: Launch the baskets

```
ASC1 - AbitMORE® SCM Commander 1.4.1  ---------------------- Row 1 to 3 of 3
Home    > Basket processing
Command => _                                                  Scroll => CSR

Basket DSNs mask: U87462.B.T.DSNADMUT_____________________

Execute: LB Launch basket  FL Filter tags and launch
Update : ED Edit basket    TR Transform basket
Review : VI View basket    BR Browse basket          LM List members
Manage : DL Delete basket  RN Rename basket

Cmd  Data set Name                                  Created     Reference  Dsorg
---  ---------------------------------------------  ----------  ---------- -----
 lb  U87462.B.T.DSNADMUT.D10083.T48198.S.ACPD       2010/03/24  2012/08/15  PS
 lb  U87462.B.T.DSNADMUT.D10083.T48198.T.ACPD       2010/03/24  2012/08/15  PS
 lb  U87462.B.T.DSNADMUT.D10083.T48198.T.BASELIB    2010/03/24  2012/08/15  PS
********************************** Bottom of data **********************************
```

Step 04: Select XML

```
ASC1 - AbitMORE® SCM Commander 1.4.1  -------------------------- Row 1 from 1
Home    > Basket processing > Launch
Command =>                                                    Scroll => CSR

Selected basket: U87462.B.T.DSNADMUT.D10083.T48198.S.ACPD

Execute: EX Execute XML service for each basket record
Review : VI View input tags   VO View output tags

Flt= *_______ *_______ *_______ *_______ ________

Cmd  Service   Scope     Message   Color     Category
---  --------  --------  --------  --------  --------
 ex  DSS       SERVICE   ALLOCATE  GREEN     DSN
********************************** Bottom of data **********************************
```

Above is an illustration of how ASC helps an XML newbee to take advantage of the extreme power of XML services. I.e. ASC "guides" you to picking (= finding !) the appropriate XML service (in this case DSS/SERVICE/ALLOCATE), and gives you all sorts of extra XML execution features (step 05, part 2/2 below) to get the job done.

Step 05: Execute XML (1/2)

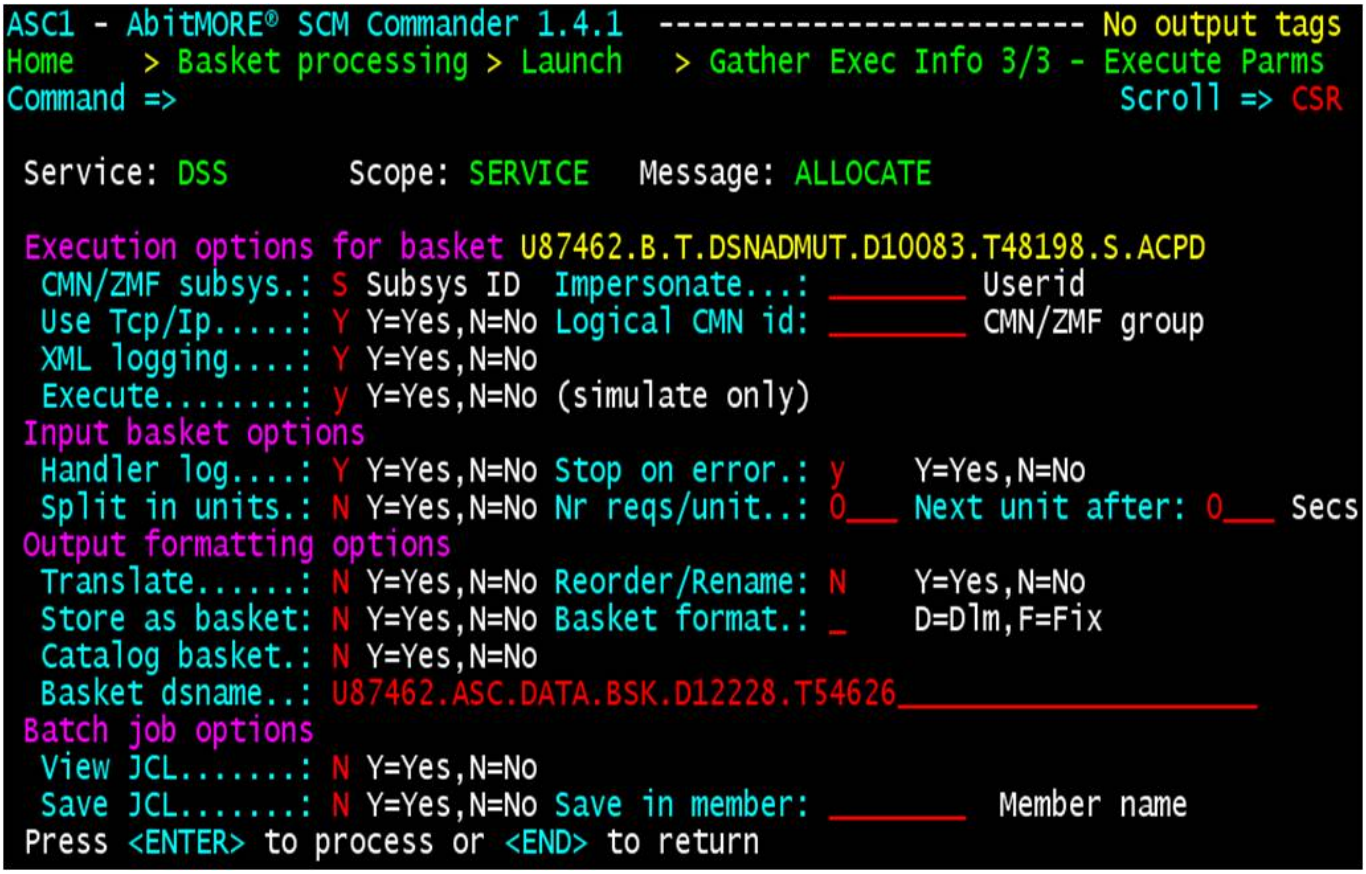
```
ASC1 - AbitMORE® SCM Commander 1.4.1  ---------------------- Row 1 to 11 of 11
Home    > Basket processing  > Launch   > Gather Exec Info 1/3 - Request tags
Command => _                                                  Scroll => CSR

Press <ENTER> to process, <END> to return

Service: DSS       Scope: SERVICE   Message: ALLOCATE

List of valid input tags     Lvl Len  Enter the value for the desired tags
---------------------------- --- ---- ----------------------------------------
mvsLib                         1  0255 STORED IN BASKET_____________________
volume                         1  0006 _____________________________________
unitName                       1  0008 STORED IN BASKET_____________________
spaceType                      1  0003 STORED IN BASKET_____________________
dataOrg                        1  0008 STORED IN BASKET_____________________
recordFormat                   1  0003 STORED IN BASKET_____________________
primarySpace                   1  0006 STORED IN BASKET_____________________
secondarySpace                 1  0006 STORED IN BASKET_____________________
dirBlocks                      1  0006 STORED IN BASKET_____________________
recordLength                   1  0006 STORED IN BASKET_____________________
blockSize                      1  0006 STORED IN BASKET_____________________
********************************** Bottom of data **********************************
```

Step 05: Execute XML (2/2)



So in this specific scenario the basket processing will trigger about 100 to 200 XML requests for each application, 1 for each record in each of the 3 basket DSNs in this scenario. Do the math if you'd have hundreds of applications (and then think of how long it would take to achieve the same result if you have to do this without using XML services ... hours, days or weeks?).

What is even (Abit)MORE: executing all these XML services can be done without "disturbing" the CMN/ZMF STC too much. This because of the **input basket options** shown above: turning on the options **Split in units**, **Nr req/unit** and **Next unit after** will facilitate a kind of work load balancing to actually execute all those services. That will avoid that regular ZMF users would experience bad response times from the CMN/ZMF started task while all those XML services are being processed (like compile jobs or promotion jobs that would take extremely long ...). Because of this, the scenario above can be executed also during CMN/ZMF primetime hours (instead of during none business hours at night or during weekends).

Source URL (retrieved on *25/04/2024 - 12:56*): http://www.abitmore-scm.com/products/commander/builtin-solutions/ascz0202