Status:	New	Priority: Should have
Author:	Andreas Dörler	Category:
Created:	2014-06-09	Assigned To:
Updated:	2014-06-09	Due date:
 Subject:	Make \$localPackagePath co	,
Description		
•		
Hi,		
,		
currently \$localP	ackagePath is hardcoded to	
	-	
\$localPackage	ePath = \$deployment->getWorkspac	ePath(\$application);
and the <b>workspa</b>	cePath is not configurable as far as	I've seen:
/ <b>*</b> *		
* Get a loca	workspace directory for the applica	tion
*/		
public function	on getWorkspacePath(Application \$a	application) {
return FLO	N PATH DATA 'Surf/' Sthis->get	Name() . '/' . \$application->getName();
	w_i ATT_DATA. Odil/ . \$tilis->geti	varie(): / : vapplication->getivarie();
}	W_i ATT_DATA. Guii/ . \$uiiis->geu	value(). / . wappincation->getivalue(),
_	W_I ATT_DATA. Sull/ . Willis >geti	value(): / : ψαρριισαίιση->getivalue(),
_	W_i ATT_DATA : Suii/ : \$\tile >get	value() . 7 . papplication->getivalue(),
}		
} If one uses Jenkir		checked out from GIT by Jenkins itself, so there is no need to do it again in
} If one uses Jenkir SURF.	ns for deployment, the source will be	
} If one uses Jenkir SURF.		
} If one uses Jenkir SURF. Currently we use	ns for deployment, the source will be the following workaround:	checked out from GIT by Jenkins itself, so there is no need to do it again in
}  If one uses Jenkin SURF.  Currently we use  \$application-	ns for deployment, the source will be the following workaround: >setOption('transferMethod', NULL)	checked out from GIT by Jenkins itself, so there is no need to do it again in
}  If one uses Jenkin SURF.  Currently we use  \$application- \$application-	ns for deployment, the source will be the following workaround: >setOption('transferMethod', NULL) >setOption('packageMethod', NULL)	checked out from GIT by Jenkins itself, so there is no need to do it again in ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout
}  If one uses Jenkir SURF.  Currently we use  \$application- \$application-	ns for deployment, the source will be the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrsync', array('customrsyn	checked out from GIT by Jenkins itself, so there is no need to do it again in ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath',
} If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin	ns for deployment, the source will be the following workaround: >setOption('transferMethod', NULL) >setOption('packageMethod', NULL >setOption('customrsync', array('customrsync', array('customrksync', array('customrksync	checked out from GIT by Jenkins itself, so there is no need to do it again in  ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e()));
} If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin	ns for deployment, the source will be the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrsync', array('customrsyn	checked out from GIT by Jenkins itself, so there is no need to do it again in  ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e()));
} If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin	ns for deployment, the source will be the following workaround: >setOption('transferMethod', NULL) >setOption('packageMethod', NULL >setOption('customrsync', array('customrsync', array('customrksync', array('customrksync	checked out from GIT by Jenkins itself, so there is no need to do it again in  ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e()));
}  If one uses Jenkin SURF.  Currently we use  \$application- \$application- '/var/lib/jenkin: \$workflow->a	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrkspace/'.\$application->getNamaddTask('typo3.surf:transfer:customr	checked out from GIT by Jenkins itself, so there is no need to do it again in ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); sync', 'transfer', \$application);
}  If one uses Jenkin SURF.  Currently we use  \$application- \$application- '/var/lib/jenkin: \$workflow->a	ns for deployment, the source will be the following workaround: >setOption('transferMethod', NULL) >setOption('packageMethod', NULL >setOption('customrsync', array('customrsync', array('customrksync', array('customrksync	checked out from GIT by Jenkins itself, so there is no need to do it again in ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); sync', 'transfer', \$application);
} If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrkspace/'.\$application->getNamaddTask('typo3.surf:transfer:customr	checked out from GIT by Jenkins itself, so there is no need to do it again in ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); rsync', 'transfer', \$application);
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('cuss/workspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set and the set	checked out from GIT by Jenkins itself, so there is no need to do it again in  ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); sync', 'transfer', \$application);  option if set: b']['customLocalPackagePath'])) ?
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('cuss/workspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set and the set	checked out from GIT by Jenkins itself, so there is no need to do it again in   ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); rsync', 'transfer', \$application);  option if set:
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('cuss/workspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set and the set	checked out from GIT by Jenkins itself, so there is no need to do it again in  ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); sync', 'transfer', \$application);  option if set: b']['customLocalPackagePath'])) ?
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('cuss/workspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set and the set	checked out from GIT by Jenkins itself, so there is no need to do it again in  ; // disable rsync ); // disable SURF git checkout - we use jenkins git checkout stomLocalPackagePath', e())); sync', 'transfer', \$application);  option if set: b']['customLocalPackagePath'])) ?
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a In a copy of the d \$localPackage \$options['cust	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrkspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set of the	checked out from GIT by Jenkins itself, so there is no need to do it again in  (i) // disable rsync (i); // disable SURF git checkout - we use jenkins git checkout (stomLocalPackagePath', (e())); (sync', 'transfer', \$application);  (poption if set: (i) ['customLocalPackagePath'])) ? (i) : \$deployment->getWorkspacePath(\$application);
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a In a copy of the d \$localPackage \$options['cust	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('cuss/workspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set and the set	checked out from GIT by Jenkins itself, so there is no need to do it again in  (i) // disable rsync (i); // disable SURF git checkout - we use jenkins git checkout (stomLocalPackagePath', (e())); (sync', 'transfer', \$application);  (poption if set: (i) ['customLocalPackagePath'])) ? (i) : \$deployment->getWorkspacePath(\$application);
If one uses Jenkin SURF. Currently we use \$application- \$application- \$application- '/var/lib/jenkin: \$workflow->a  In a copy of the d \$localPackage \$options['cust	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrkspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set of the	checked out from GIT by Jenkins itself, so there is no need to do it again in  (*, // disable rsync*); // disable SURF git checkout - we use jenkins git checkout (*stomLocalPackagePath', (*e())); (*sync', 'transfer', \$application);  (*ption if set: (*) ['customLocalPackagePath'])) ?  (*) ['customLocalPackagePath'])) ?  (*) ['s \$deployment->getWorkspacePath(\$application);
If one uses Jenkin SURF. Currently we use \$application- \$application- '/var/lib/jenkin: \$workflow->a In a copy of the d \$localPackage \$options['cust	the following workaround:  >setOption('transferMethod', NULL) >setOption('packageMethod', NULL) >setOption('packageMethod', NULL) >setOption('customrsync', array('customrkspace/'.\$application->getNamaddTask('typo3.surf:transfer:customrefault RsyncTask we use the given of the set of the	checked out from GIT by Jenkins itself, so there is no need to do it again in  (*,"/ disable rsync* (*); // disable SURF git checkout - we use jenkins git checkout (*); // disable SURF git checkout - we use jenkins git checkout (*); // disable SURF git checkout - we use jenkins git checkout (*); // disable SURF git checkout - we use jenkins git checkout (*); // disable SURF git checkout - we use jenkins git checkout (*); // disable surprise for the checkout (*); // disable surprise for the checkout - we use jenkins git checkout (*); // disable surprise for the checkout

## History

2015-08-04 1/1