NEWSCYCLE SOLUTIONS

Digital Tags Manual

Release 7.4
November 17, 2014

NEWSCYCLE Solutions — Confidential & Proprietary

Digital 7.4 Tags Manual

© 2014 NEWSCYCLE Solutions.

Any and/or all sections of this manual may be duplicated for training purposes.

Prior permission is not needed from NEWSCYCLE Solutions.

This guide and the software described in it are copyrighted material. All rights are reserved.

NEWSCYCLE Solutions
7900 International Drive
Suite 800
Bloomington, MN 55425

REV NOVEMBER 17, 2014

1. Tags Manual 16
1.1 PDF Version 17
1.2 Tag Index 18
1.3 Object File Index 22
1.4 Tag Parsing Order 31
1.5 HTML and Includes 36
1.5.1 Document structure 37
1.5.2 Fonts 38
1.5.3 Meta tags 41
1.5.3.1 Article90.pbo 42
1.5.3.2 ArticleHead90.pbo 42
1.5.3.3 ArticleHead.pbo meta tag 42
1.5.4 pbs filecheck 42
1.5.5 pbs getcontent 43
1.5.6 pbs include 44
1.5.7 Server side includes 45
1.5.8 Special graphics directory 47
1.6 Templates 48
1.6.1 Export templates 49
1.6.1.1 article_xml_asset.pbo 49
1.6.1.2 article_xml_connection.pbo 49
1.6.1.3 article_xml_paragraph.pbo 49
1.6.1.4 article_xml.pbo 50
1.6.1.4.1 article_xml_accesscontrolsitem.pbo 53
1.6.1.5 templates_Sms_xml 53
1.6.2 Folder structure 55
1.6.3 Setting MIME Types 55
1.6.4 Special templates 56
1.6.5 Template mapping 56
1.6.6 Template names and URLs 58
1.6.7 Zones 60
1.7 Object Files 61
1.7.1 About Object Files 62
1.7.2 Article classes 68
1.7.3 Global function 71
1.7.4 Global variables 72
1.7.5 Inline objects 73
1.7.6 Insert Header and Footer 73
1.7.6.1 InsertHeader.pbo InsertFooter.pbo 74
1.7.6.1.1 InsertByline.pbo 75
1.7.6.1.2 InsertBylineItem.pbo 75
1.7.6.1.3 InsertCCETypes.pbo 75
1.7.6.1.4 InsertCCETypesItem.pbo 76
1.7.6.1.5 InsertContextCategory.pbo 76
1.7.6.1.6 InsertContextProfile.pbo 76
1.7.6.1.7 InsertSessionInfo.pbo 76
1.7.6.1.8 InsertTaxonomyList.pbo 76
1.7.6.1.9 InsertTaxonomyListItem.pbo 76
1.7.7 Localdata.ini 76
1.7.8 Object script 77
1.7.9 Odd and Even variable picture handling 94
1.7.10 pbs macro 96

1.7.10.1 Election2008.pbo . . . 100
1.7.11 Position-based design . . . 100
1.7.12 Session function . . . 100
1.7.13 System variables . . . 101
1.8 Menu Tags . . . 102
1.8.1 pbs categorytree . . . 103
1.8.1.1 CategoryTree.pbo . . . 104
1.8.1.2 CategoryTreeItemActive.pbo . . . 104
1.8.1.3 CategoryTreeItemPassive.pbo . . . 104
1.8.1.4 CategoryTreeItemSeparatorColl.pbo . . . 105
1.8.1.5 CategoryTreeItemSeparatorExp.pbo . . . 105
1.8.1.6 CategoryTreeItemSeparatorNone.pbo . . . 105
1.8.1.7 CategoryTreeItemSubTreeIndention.pbo . . . 105
1.8.2 pbs ft . . . 105
1.8.3 pbs leftmenu . . . 106
1.8.4 pbs menu-image . . . 107
1.8.5 pbs menu-text . . . 107
1.8.6 pbs profiletree . . . 107
1.8.6.1 ProfileTree.pbo . . . 108
1.8.6.2 ProfileTreeItem.pbo . . . 108
1.8.6.3 ProfileTreeItemActive.pbo . . . 108
1.8.6.4 ProfileTreeItemPassive.pbo . . . 108
1.8.6.5 ProfileTreeItemSeparatorColl.pbo . . . 108
1.8.6.6 ProfileTreeItemSeparatorExp.pbo . . . 108
1.8.7 pbs qwiknavsearch . . . 109
1.8.7.1 QwikNavSearchForm.pbo . . . 109
1.9 Section Front Tags . . . 109
1.9.1 pbs oversiktnotiser . . . 110
1.9.2 pbs sectionfront . . . 110
1.10 Article Tags . . . 113
1.10.1 pbs article . . . 114
1.10.1.1 Article.pbo . . . 125
1.10.1.1.1 ArticlePhotoGalleryThumbnailItems.pbo . . . 131
1.10.1.2 ArticleAccessControls.pbo . . . 131
1.10.1.3 ArticleAccessControlsItem.pbo . . . 131
1.10.1.4 ArticleByline.pbo . . . 131
1.10.1.5 ArticleConnectionItem.pbo . . . 134
1.10.1.6 ArticleConnections.pbo . . . 135
1.10.1.7 ArticleExtLogin.pbo . . . 135
1.10.1.8 ArticleExtraPictureItem.pbo . . . 135
1.10.1.9 ArticleExtraPictures.pbo . . . 136
1.10.1.10 ArticleFacts.pbo . . . 136
1.10.1.11 ArticleForumConnectionItem.pbo . . . 137
1.10.1.12 ArticleForumConnections.pbo . . . 137
1.10.1.13 ArticleGalleryConnectionItem.pbo . . . 137
1.10.1.14 ArticleGalleryConnections.pbo . . . 137
1.10.1.15 ArticleGeoDataItem.pbo . . . 137
1.10.1.16 ArticleGeoDataItems.pbo . . . 138
1.10.1.17 ArticleHead.pbo . . . 138
1.10.1.18 ArticleHotLinks.pbo . . . 138
1.10.1.19 ArticleLinkItem.pbo . . . 139
1.10.1.20 ArticleLinks.pbo . . . 139

1.10.1.21 ArticleLocation.pbo . . . 139
1.10.1.22 ArticleModified.pbo . . . 139
1.10.1.23 ArticleMultiMedia.pbo . . . 140
1.10.1.24 ArticleMultiMediaExtension.pbo . . . 140
1.10.1.25 ArticleMultimediaLinkItem.pbo . . . 140
1.10.1.26 ArticleMultiMediaList.pbo . . . 141
1.10.1.27 ArticleMultiMediaListItem.pbo . . . 141
1.10.1.28 ArticleMultimediaListItemExtension.pbo . . . 141
1.10.1.29 ArticleMultimediaListItemXXX.pbo . . . 141
1.10.1.30 ArticleMultiMediaPlaylistItem.pbo . . . 142
1.10.1.31 ArticleParagraph.pbo . . . 142
1.10.1.32 ArticlePhotoGalleryThumbnailItem.pbo . . . 143
1.10.1.33 ArticlePhotoGalleryThumbnails.pbo . . . 143
1.10.1.34 ArticlePicture.pbo . . . 143
1.10.1.35 ArticlePublished.pbo . . . 145
1.10.1.36 ArticleSummary.pbo . . . 145
1.10.1.37 ArticleTaxonomy.pbo . . . 145
1.10.1.38 ArticleTaxonomyItem.pbo . . . 145
1.10.1.39 ArticleTaxonomyItemPathItem.pbo . . . 146
1.10.1.40 InvalidArticle.pbo . . . 146
1.10.1.41 Tips.pbo and articles . . . 146
1.10.2 pbs articledate . . . 146
1.10.3 pbs articlefield . . . 146
1.10.4 pbs article-links . . . 147
1.10.5 pbs articlemultimedia . . . 147
1.10.6 pbs artikkel-urls . . . 147
1.10.7 pbs keywordlinks . . . 147
1.10.7.1 KeywordLinkItem.pbo . . . 147
1.10.7.2 KeywordLinks.pbo . . . 148
1.10.8 pbs photogallery-links . . . 148
1.10.9 pbs themelinks . . . 148
1.10.9.1 ThemeLinkItem.pbo . . . 149
1.10.9.2 ThemeLinks.pbo . . . 149
1.10.10 Other links . . . 149
1.11 Overview Tags . . . 149
1.11.1 pbs agent . . . 151
1.11.1.1 Agent.pbo . . . 156
1.11.1.2 AgentItem.pbo . . . 156
1.11.2 pbs agentcategory . . . 157
1.11.3 pbs contentlist . . . 157
1.11.3.1 ContentList.pbo . . . 166
1.11.3.2 ContentListByline.pbo . . . 170
1.11.3.3 ContentListFacetItem.pbo . . . 170
1.11.3.4 ContentListFacetList.pbo . . . 170
1.11.3.5 ContentListFilterItem.pbo . . . 170
1.11.3.6 ContentListFilterList.pbo . . . 170
1.11.3.7 ContentListItem.pbo . . . 171
1.11.3.8 ContentListItemAccessControls.pbo . . . 175
1.11.3.9 ContentListItemAccessControlsItem.pbo . . . 176
1.11.3.10 ContentListItemImageItem.pbo . . . 176
1.11.3.11 ContentListItemTaxonomy.pbo . . . 177
1.11.3.12 ContentListItemTaxonomyItem.pbo . . . 177

1.11.3.13 ContentListItemTaxonomyItemPathItem.pbo 177
1.11.4 pbs contentstats 178
1.11.4.1 ContentStats.pbo 181
1.11.4.2 ContentStatsItem.pbo 181
1.11.5 pbs freestyle 181
1.11.5.1 FreestyleStoryArticleConnectionItem.pbo 183
1.11.5.2 FreestyleStoryItem.pbo 184
1.11.5.3 FreestyleStoryItemByline.pbo 189
1.11.5.4 FreestyleStoryItemConnectionItem.pbo 189
1.11.5.5 FreestyleStoryItemExternalLinkItem.pbo 190
1.11.5.6 FreestyleStoryItemExternalLinks.pbo 190
1.11.5.7 FreestyleStoryItemForumConnectionItem.pbo 190
1.11.5.8 FreestyleStoryItemGalleryConnectionItem.pbo 190
1.11.5.9 FreestyleStoryItemInternalLinks.pbo 190
1.11.5.10 FreestyleStoryItemKeywordLinkItem.pbo 191
1.11.5.11 FreestyleStoryItemKeywordLinks.pbo 191
1.11.5.12 FreestyleStoryItemThemeLinkItem.pbo 191
1.11.5.13 FreestyleStoryItemThemeLinks.pbo 191
1.11.5.14 FreestyleStoryList.pbo 191
1.11.6 pbs history 191
1.11.6.1 History.pbo 192
1.11.6.2 HistoryItem.pbo 192
1.11.7 pbs keywordindex 192
1.11.7.1 KeywordIndexHeader.pbo 193
1.11.7.2 KeywordIndexLink.pbo 193
1.11.7.3 KeywordIndexTitle.pbo 193
1.11.8 pbs mostread 194
1.11.8.1 MostRead.pbo 196
1.11.8.2 MostReadItem.pbo 196
1.11.8.3 MostReadItemAccessControls.pbo 197
1.11.8.4 MostReadItemAccessControlsItem.pbo 197
1.11.8.5 MostReadItemTaxonomy.pbo 197
1.11.8.6 MostReadItemTaxonomyItem.pbo 197
1.11.8.7 MostReadItemTaxonomyItemPathItem.pbo 198
1.11.9 pbs newslist 198
1.11.9.1 News.pbo 217
1.11.9.2 NewsItem.pbo 217
1.11.9.3 NewsItemAccessControls.pbo 223
1.11.9.4 NewsItemAccessControlsItem.pbo 224
1.11.9.5 NewsItemArticleConnectionItem.pbo 224
1.11.9.6 NewsItemByline.pbo 224
1.11.9.7 NewsItemConnectionItem.pbo 226
1.11.9.8 NewsItemExternalLinkItem.pbo 227
1.11.9.9 NewsItemExternalLinks.pbo 227
1.11.9.10 NewsItemExtraPictureItem.pbo 228
1.11.9.11 NewsItemExtraPictures.pbo 229
1.11.9.12 NewsItemForumConnectionItem.pbo 230
1.11.9.13 NewsItemForumGroupLink.pbo 230
1.11.9.14 NewsItemGalleryConnectionItem.pbo 230
1.11.9.15 NewsItemGeoData.pbo 230
1.11.9.16 NewsItemGeoDataItem.pbo 230
1.11.9.17 NewsItemInternalLinks.pbo 230

1.11.9.18 NewsItemKeywordLinkItem.pbo . . . 230
1.11.9.19 NewsItemKeywordLinks.pbo . . . 230
1.11.9.20 NewsItemMultiMediaExtension.pbo . . . 231
1.11.9.21 NewsItemParagraphItem.pbo . . . 231
1.11.9.22 NewsItemTaxonomy.pbo . . . 232
1.11.9.23 NewsItemTaxonomyItem.pbo . . . 232
1.11.9.24 NewsItemTaxonomyItemPathItem.pbo . . . 232
1.11.9.25 NewsItemTaxonomyWordItem.pbo . . . 232
1.11.9.26 NewsItemTaxonomyWords.pbo . . . 232
1.11.9.27 NewsItemThemeLinkItem.pbo . . . 232
1.11.9.28 NewsItemThemeLinks.pbo . . . 233
1.11.9.29 RelStoriesItem.pbo . . . 233
1.11.9.30 RelStoriesItems.pbo . . . 233
1.11.9.31 URL Parameters and overview tags . . . 233
1.11.10 pbs related . . . 234
1.11.10.1 ThemeLinkItem.pbo and overview tags . . . 234
1.11.10.2 ThemeLinks.pbo and pbs related . . . 235
1.11.11 pbs subcategories . . . 235
1.11.11.1 SubcatSimpleLink.pbo . . . 236
1.12 Images . . . 236
1.12.1 Access control for images . . . 237
1.12.1.1 PhotoGalleryExtLogin.pbo . . . 237
1.12.2 Story-linked photo gallery . . . 237
1.12.3 Generate graphical text . . . 238
1.12.4 Order picture . . . 241
1.12.5 pbs link and images . . . 243
1.12.6 pbs photogallery . . . 244
1.12.6.1 ArticleConnectionItem.pbo and images . . . 245
1.12.6.2 ArticleConnections.pbo and images . . . 245
1.12.6.3 PhotoGallery.pbo . . . 245
1.12.6.4 PhotoGalleryAccessControls.pbo . . . 247
1.12.6.5 PhotoGalleryAccessControlsItem.pbo . . . 247
1.12.6.6 PhotoGalleryItemExtraFieldItem.pbo . . . 248
1.12.6.7 PhotoGalleryItemExtraFields.pbo . . . 248
1.12.6.8 PhotoGalleryItemTaxonomy.pbo . . . 248
1.12.6.9 PhotoGalleryItemTaxonomyItem.pbo . . . 248
1.12.6.10 PhotoGalleryTaxonomy.pbo . . . 248
1.12.6.11 PhotoGalleryTaxonomyItem.pbo . . . 248
1.12.6.12 PhotoGalleryTaxonomyItemPathItem.pbo . . . 248
1.12.7 pbs photogallery-latest . . . 249
1.12.7.1 PhotoGalleryLatest.pbo . . . 253
1.12.7.2 PhotoGalleryLatestItem.pbo . . . 253
1.12.7.3 PhotoGalleryLatestItemAccessControls.pbo . . . 255
1.12.7.4 PhotoGalleryLatestItemAccessControlsItem.pbo . . . 255
1.12.7.5 PhotoGalleryLatestItemExtraFieldItem.pbo . . . 255
1.12.7.6 PhotoGalleryLatestItemExtraFields.pbo . . . 255
1.12.7.7 PhotoGalleryLatestItemListTaxonomy.pbo . . . 256
1.12.7.8 PhotoGalleryLatestItemListTaxonomyItem.pbo . . . 256
1.12.7.9 PhotoGalleryLatestItemListTaxonomyItemPathItem.pbo . . . 256
1.12.7.10 PhotoGalleryLatestItemTaxonomy.pbo . . . 256
1.12.7.11 PhotoGalleryLatestItemTaxonomyItem.pbo . . . 256
1.12.8 pbs photogallery-thumbnails . . . 256

1.12.8.1 PhotoGalleryThumbnailItem.pbo . . . 257
1.12.8.2 PhotoGalleryThumbnails.pbo . . . 258
1.12.8.3 PhotoGalleryThumbnailsAccessControls.pbo . . . 259
1.12.8.4 PhotoGalleryThumbnailsAccessControlsItem.pbo . . . 259
1.12.9 pbs ptshowimage . . . 259
1.12.9.1 ShowImage.pbo . . . 260
1.12.10 Photo album export . . . 261
1.12.10.1 ExpPhotoAlbum.pbo . . . 262
1.12.10.1.1 ExpPhotoAlbumAccessControlsItem.pbo . . . 262
1.12.10.2 ExpPhotoAlbumItem.pbo . . . 263
1.12.10.3 ExpPhotoAlbums.pbo . . . 263
1.12.11 Set picture as desktop background . . . 263
1.12.12 Watermark images . . . 264
1.12.13 pbs ptimages . . . 265
1.12.13.1 ImageItem.pbo . . . 268
1.12.13.2 ImageItemRowEnd.pbo . . . 268
1.12.13.3 ImageItemRowStart.pbo . . . 268
1.13 Newsletter Tags . . . 268
1.13.1 Setup newsletters . . . 269
1.13.2 Web based management for newsletters . . . 270
1.13.2.1 SubcatSimpleLink1.pbo . . . 273
1.13.3 Newsletter template . . . 274
1.13.3.1 Newsletter.txt . . . 274
1.13.3.2 NewsletterAd.txt . . . 274
1.13.3.3 NewsletterArticle.txt . . . 274
1.13.3.4 NewsletterCategory.txt . . . 275
1.13.3.5 NewsletterLinks.txt . . . 275
1.13.3.6 NewsletterOAMSAdHTML.pbo . . . 275
1.13.3.7 NewsletterOAMSAdSpot.txt . . . 275
1.13.3.8 NewsletterSniff.txt . . . 276
1.13.4 pbs Newslad . . . 276
1.13.5 pbs newsletter_subscriptionform . . . 276
1.13.5.1 NewsletterActivated.pbo . . . 277
1.13.5.2 NewsLetterActivatedError.pbo . . . 277
1.13.5.3 NewsletterRemoved.pbo . . . 277
1.13.5.4 NewsLetterRemoveError.pbo . . . 277
1.13.5.5 NewsLetterRemoveMail.pbo . . . 277
1.13.5.6 NewsLetterSubscribeMail.pbo . . . 277
1.13.5.7 NewsLetterSubscriptionForm.pbo . . . 277
1.13.6 Subscription page with Sniff Search . . . 279
1.14 Searching . . . 280
1.14.1 Image search . . . 281
1.14.1.1 pbs imagesearchresult . . . 282
1.14.1.1.1 SearchResultImage.pbo . . . 283
1.14.1.1.2 SearchResultImageItem.pbo . . . 283
1.14.1.1.3 SearchResultImageRow.pbo . . . 284
1.14.1.1.4 ImageSearchError.pbo . . . 284
1.14.1.1.5 SearchResultImageItemAccessControlsItem.pbo . . . 284
1.14.1.1.6 SearchResultImageItemAccessControls.pbo . . . 285
1.14.2 Result page . . . 285
1.14.2.1 Building the page without object files . . . 285
1.14.2.1.1 pbs search_criteria . . . 285

1.14.2.1.2 pbs search_hits . . . 285
1.14.2.1.3 pbs search_next_link . . . 285
1.14.2.1.4 pbs search_previous_link . . . 285
1.14.2.1.5 pbs search_totalhits . . . 285
1.14.2.1.6 pbs searchresult without object files . . . 285
1.14.2.2 pbs searchresult . . . 286
1.14.2.2.1 StorySearchNavigatorItem.pbo . . . 288
1.14.2.2.2 ImageSearchNavigator.pbo . . . 288
1.14.2.2.3 ImageSearchNavigatorItem.pbo . . . 289
1.14.2.2.4 SearchError.pbo . . . 289
1.14.2.2.5 SearchNavigator.pbo . . . 289
1.14.2.2.6 SearchNavigatorItem.pbo . . . 289
1.14.2.2.7 SearchResult.pbo . . . 289
1.14.2.2.8 SearchResultItem.pbo . . . 292
1.14.2.2.9 StorySearchNavigator.pbo . . . 293
1.14.2.2.10 SearchNavigatorFilter.pbo . . . 293
1.14.2.2.11 SearchNavigatorFilterItem.pbo . . . 293
1.14.2.2.12 SearchResultItemByline.pbo . . . 294
1.14.2.2.13 SearchResultItemTaxonomyItemPathItem.pbo . . . 296
1.14.2.2.14 SearchResultItemTaxonomy.pbo . . . 296
1.14.2.2.15 SearchResultItemTaxonomyItem.pbo . . . 296
1.14.2.2.16 SearchResultItemAccessControlsItem.pbo . . . 296
1.14.2.2.17 SearchResultItemAccessControls.pbo . . . 297
1.14.3 Search form . . . 297
1.14.4 Search other websites . . . 301
1.14.5 Search syntax . . . 303
1.15 Custom Content Engine . . . 304
1.15.1 CCE setup . . . 306
1.15.2 CCE data retrieval tags . . . 306
1.15.3 CCE search form . . . 307
1.15.4 CCE Export . . . 310
1.15.4.1 CCEExport.pbo . . . 310
1.15.4.2 CCEExportCustomFieldItem.pbo . . . 310
1.15.4.3 CCEExportDateMapItem.pbo . . . 310
1.15.4.4 CCEExportGroupBy.pbo . . . 310
1.15.4.5 CCEExportImageItem.pbo . . . 311
1.15.4.6 CCEExportItem.pbo . . . 311
1.15.4.6.1 CCEExportItemAccessControlsItem.pbo . . . 313
1.15.4.7 CCEExportItemTaxonomywords.pbo . . . 313
1.15.4.8 CCEExportItemTaxonomywordsItem.pbo . . . 313
1.15.4.9 CCEExportItemTaxonomywordsItemPathItem.pbo . . . 314
1.15.5 CCE tip a friend . . . 314
1.15.6 pbs cce . . . 314
1.15.6.1 CCE_ModuleShortName.pbo . . . 329
1.15.6.2 CCE_ModuleShortNameGroupHeader.pbo . . . 331
1.15.6.3 CCE_ModuleShortNameItem.pbo . . . 332
1.15.6.4 CCE_ModuleShortNameItemAccessControls.pbo . . . 336
1.15.6.5 CCE_ModuleShortNameItemAccessControlsItem.pbo . . . 336
1.15.6.6 CCE_ModuleShortNameItemEventDateItem.pbo . . . 337
1.15.6.7 CCE_ModuleShortNameItemTaxonomy.pbo . . . 337
1.15.6.8 CCE_ModuleShortNameItemTaxonomyItem.pbo . . . 337
1.15.6.9 CCE_ModuleShortNameItemTaxonomyItemPathItem.pbo . . . 338

1.15.6.10 CCE_ModuleShortNameNavigator.pbo . . . 338
1.15.6.11 CCE_ModuleShortNameNavigatorFilter.pbo . . . 338
1.15.6.12 CCE_ModuleShortNameNavigatorFilterItem.pbo . . . 338
1.15.6.13 CCE_ModuleShortNameNavigatorItem.pbo . . . 339
1.15.6.14 CCE_ModuleShortNamePageLink . . . 339
1.15.6.15 CCE_ModuleShortNamePageLinkItem . . . 339
1.15.6.16 CCE_MPLACE.pbo . . . 340
1.15.6.17 CCE_MPLACEitem.pbo . . . 340
1.15.6.18 CCE-search_ovr.pbs . . . 340
1.15.7 pbs ccedetails . . . 340
1.15.7.1 CCE_ModuleShortNameDetails.pbo . . . 341
1.15.7.2 CCE_ModuleShortNameDetailsAccessControls.pbo . . . 345
1.15.7.3 CCE_ModuleShortNameDetailsAccessControlsItem.pbo . . . 345
1.15.7.4 CCE_ModuleShortNameDetailsEventDateListItem.pbo . . . 345
1.15.7.5 CCE_ModuleShortNameDetailsTaxonomy.pbo . . . 346
1.15.7.6 CCE_ModuleShortNameDetailsTaxonomyItem.pbo . . . 346
1.15.7.7 CCE_ModuleShortNameDetailsTaxonomyItemPathItem.pbo . . . 346
1.15.7.8 CCE_ModuleShortNamePrintDetail.pbo . . . 346
1.15.7.9 CCE_ModuleShortNameRelated.pbo . . . 346
1.15.8 pbs ccefields . . . 347
1.15.8.1 CCE_ModuleShortNameFieldsFieldName.pbo . . . 350
1.15.8.2 CCE_ModuleShortNameFieldsFieldNameItem.pbo . . . 350
1.15.8.3 CCE_ModuleShortNameFieldsFieldNameRow.pbo . . . 351
1.15.9 pbs LastUpdate for CCE . . . 351
1.15.10 pbs usersubmitform . . . 351
1.16 User Generated Content Tags . . . 352
1.16.1 User Submitted Stories . . . 355
1.16.2 pbs userarticleform . . . 355
1.16.2.1 ArticleForm.pbo . . . 356
1.16.2.2 ArticleFormMediaItem.pbo . . . 359
1.16.2.3 ArticleFormSectionItem.pbo . . . 359
1.16.2.4 UserSubArticles_ArticleFormNotLoggedIn . . . 360
1.16.3 pbs usersubmitform for UGC . . . 360
1.16.3.1 UserSubmitForm.pbo . . . 362
1.16.3.2 UserSubmitFormArticle.pbo . . . 368
1.16.3.3 UserSubmitFormArticleCategoryList.pbo . . . 368
1.16.3.3.1 UserSubmitFormArticleCategoryListItem.pbo . . . 369
1.16.3.4 UserSubmitFormArticleErrorFailed.pbo . . . 369
1.16.3.5 UserSubmitFormArticleErrorInvalidInput.pbo . . . 369
1.16.3.6 UserSubmitFormArticleErrorNotCreator.pbo . . . 369
1.16.3.7 UserSubmitFormArticleErrorNotLoggedIn.pbo . . . 369
1.16.3.8 UserSubmitFormArticleErrorNoUGCAccess.pbo . . . 369
1.16.3.9 UserSubmitFormArticleExtraFields.pbo . . . 369
1.16.3.9.1 UserSubmitFormArticleExtraFieldsItem.pbo . . . 369
1.16.3.10 UserSubmitFormArticleMediaItem.pbo . . . 369
1.16.3.11 UserSubmitFormArticleSectionItem.pbo . . . 369
1.16.3.12 UserSubmitFormArticleTaxonomyList.pbo . . . 370
1.16.3.12.1 UserSubmitFormArticleTaxonomyListItem.pbo . . . 370
1.16.3.13 UserSubmitFormCategoryList.pbo . . . 370
1.16.3.13.1 UserSubmitFormCategoryListItem.pbo . . . 370
1.16.3.14 UserSubmitFormCCE.pbo . . . 370
1.16.3.15 UserSubmitFormCCECategoryList.pbo . . . 370

1.16.3.15.1 UserSubmitFormCCECategoryListItem.pbo . . . 371
1.16.3.16 UserSubmitFormCCEErrorFailed.pbo . . . 371
1.16.3.17 UserSubmitFormCCEErrorInvalidInput.pbo . . . 371
1.16.3.18 UserSubmitFormCCEErrorNotCreator.pbo . . . 371
1.16.3.19 UserSubmitFormCCEErrorNotLoggedIn.pbo . . . 371
1.16.3.20 UserSubmitFormCCEErrorNoUGCAccess.pbo . . . 371
1.16.3.21 UserSubmitFormCCEFieldList.pbo . . . 371
1.16.3.21.1 UserSubmitFormCCEFieldListItem.pbo . . . 371
1.16.3.22 UserSubmitFormCCEModule.pbo . . . 371
1.16.3.23 UserSubmitFormCCERecurrencePattern.pbo . . . 372
1.16.3.24 UserSubmitFormCCETaxonomyList.pbo . . . 375
1.16.3.24.1 UserSubmitFormCCETaxonomyListItem.pbo . . . 375
1.16.3.25 UserSubmitFormErrorFailed.pbo . . . 375
1.16.3.26 UserSubmitFormErrorInvalidInput.pbo . . . 376
1.16.3.27 UserSubmitFormErrorNotCreator.pbo . . . 376
1.16.3.28 UserSubmitFormErrorNotLoggedIn.pbo . . . 376
1.16.3.29 UserSubmitFormErrorNoUGCAccess.pbo . . . 376
1.16.3.30 UserSubmitFormExtraFields.pbo . . . 376
1.16.3.30.1 UserSubmitFormExtraFieldsItem.pbo . . . 376
1.16.3.31 UserSubmitFormPhotoGallery.pbo . . . 377
1.16.3.32 UserSubmitFormPhotoGalleryCategoryList.pbo . . . 377
1.16.3.32.1 UserSubmitFormPhotoGalleryCategoryListItem.pbo . . . 377
1.16.3.33 UserSubmitFormPhotoGalleryErrorFailed.pbo . . . 377
1.16.3.34 UserSubmitFormPhotoGalleryErrorInvalidInput.pbo . . . 377
1.16.3.35 UserSubmitFormPhotoGalleryErrorNotCreator.pbo . . . 377
1.16.3.36 UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo . . . 377
1.16.3.37 UserSubmitFormPhotoGalleryErrorNoUGCAccess.pbo . . . 377
1.16.3.38 UserSubmitFormPhotoGalleryExtraFields.pbo . . . 377
1.16.3.38.1 UserSubmitFormPhotoGalleryExtraFieldsItem.pbo . . . 377
1.16.3.39 UserSubmitFormPhotoGalleryPhotoItem.pbo . . . 377
1.16.3.40 UserSubmitFormPhotoGalleryTaxonomyList.pbo . . . 377
1.16.3.40.1 UserSubmitFormPhotoGalleryTaxonomyListItem.pbo . . . 378
1.16.3.41 UserSubmitFormTaxonomyList.pbo . . . 378
1.16.3.41.1 UserSubmitFormTaxonomyListItem.pbo . . . 378
1.17 User Comments . . . 378
1.17.1 pbs UserComments . . . 379
1.17.1.1 UserComments.pbo . . . 381
1.17.1.2 UserCommentsItem.pbo . . . 382
1.17.2 pbs UserCommentsSubmit . . . 384
1.17.2.1 UserCommentsSubmit.pbo . . . 385
1.17.2.2 UserCommentsSubmitError.pbo . . . 386
1.18 Q&A Tags . . . 387
1.18.1 pbs pollarchive . . . 388
1.18.1.1 PollArchiveItem.pbo . . . 389
1.18.2 pbs polldata . . . 390
1.18.2.1 PollDataResults.pbo . . . 391
1.18.2.2 PollDataResultsItem.pbo . . . 391
1.18.2.3 PollDataResultsItemEntry.pbo . . . 392
1.18.2.4 PollForm.pbo . . . 392
1.18.2.5 PollFormAlternatives.pbo . . . 393
1.18.2.6 PollFormAlternativesItem.pbo . . . 393
1.18.2.7 PollFormItem.pbo . . . 393

1.18.2.8 Polls_Error.pbo with pbs polldata . . . 393
1.18.2.9 Polls_Saved.pbo with pbs polldata . . . 393
1.18.2.10 Poll Statistics . . . 393
1.18.3 pbs polllist . . . 393
1.18.3.1 PollList.pbo . . . 394
1.18.3.2 PollListItem.pbo . . . 394
1.18.4 pbs pollmini . . . 395
1.18.4.1 Pollmini.pbo . . . 396
1.18.4.2 PollMiniChart.pbo . . . 397
1.18.4.3 PollMiniNoPoll.pbo . . . 397
1.18.4.4 PollMiniQuery.pbo . . . 397
1.18.5 pbs pollquestion . . . 398
1.18.6 pbs pollsave . . . 398
1.18.6.1 Polls_Error.pbo with pbs questform . . . 398
1.18.6.2 Polls_Saved.pbo with pbs questform . . . 398
1.18.7 pbs quest_multipage . . . 399
1.18.7.1 MultiPageQuestIndex.pbo . . . 399
1.18.7.2 MultiPageQuestPage.pbo . . . 399
1.18.7.3 MultiPageQuestPageX.pbo . . . 400
1.18.7.4 MultiPageQuestSubmit.pbo . . . 400
1.18.7.5 MultiPageQuestWrongAnswer.pbo . . . 401
1.18.7.6 Multipage self test . . . 401
1.18.7.7 MultiQuestAltItem.pbo . . . 401
1.18.7.8 Poll_Error.pbo with pbs quest_multipage . . . 402
1.18.7.9 Polls_Saved.pbo with pbs quest_multipage . . . 402
1.18.8 pbs questform . . . 402
1.18.8.1 QuestForm.pbo . . . 402
1.18.8.2 QuestFormAlternativesItem.pbo . . . 403
1.18.8.3 QuestFormItem.pbo . . . 403
1.19 Banner Ads . . . 404
1.19.1 pbs ad . . . 408
1.19.1.1 AdSpot.pbo . . . 408
1.19.1.2 FlashBanner.pbo . . . 409
1.19.1.3 LayerAdSpot.pbo . . . 410
1.19.1.4 OamsAdSpot.pbo . . . 410
1.19.2 SAXOTECH Online ads on other sites . . . 410
1.20 Links . . . 412
1.20.1 pbs links . . . 413
1.20.2 pbs linkscategorylist . . . 415
1.20.2.1 Links.pbo . . . 416
1.20.2.2 LinksItem.pbo . . . 417
1.20.2.3 LinksSubcat.pbo . . . 417
1.20.2.4 Links vote form . . . 418
1.21 Page Designer Tags . . . 419
1.21.1 pbs contentbox . . . 420
1.22 Net Guest Tags . . . 420
1.22.1 NetGuest.pbo . . . 421
1.22.2 pbs netguest . . . 421
1.22.2.1 NetGuestQuestionItem.pbo . . . 422
1.22.3 pbs netguest_list . . . 422
1.22.3.1 NetGuestList.pbo . . . 423
1.22.3.2 NetGuestListItem.pbo . . . 423

1.22.4 pbs netguest_questionform . . . 424
1.22.4.1 NetGuestQuestionForm.pbo . . . 424
1.23 Subscriptions . . . 425
1.23.1 Login page . . . 426
1.23.1.1 pbs login . . . 426
1.23.1.2 pbs logininfo . . . 427
1.23.1.3 pbs target . . . 427
1.23.2 pbs changepassword . . . 427
1.23.3 pbs global . . . 429
1.23.4 pbs subscriberdata . . . 429
1.23.4.1 SubscriberData.pbo . . . 430
1.23.5 pbs userid . . . 430
1.23.6 Self-created users . . . 430
1.24 User Registration . . . 433
1.24.1 Setup user registration . . . 434
1.24.2 Advanced login . . . 440
1.24.2.1 pbs ureg_email . . . 443
1.24.2.2 pbs ureg_message . . . 443
1.24.3 Edit user registration . . . 444
1.24.4 Log out . . . 445
1.24.5 Lost password . . . 446
1.24.5.1 PwdSent.pbo . . . 447
1.24.6 pbs adregusertargets . . . 447
1.24.7 pbs extra_income . . . 448
1.24.8 pbs reguserinfo . . . 449
1.24.8.1 Userreginfo.pbo . . . 449
1.25 Staff Listing . . . 450
1.25.1 pbs departmentlist . . . 451
1.25.1.1 DepList.pbo . . . 451
1.25.1.2 DepListItem.pbo . . . 451
1.25.2 pbs departmentoverview . . . 451
1.25.2.1 DepOverview.pbo . . . 452
1.25.2.2 DepOverviewGroupHeader.pbo . . . 453
1.25.2.3 DepOverviewHeader.pbo . . . 453
1.25.2.4 DepOverviewItem.pbo . . . 453
1.25.3 pbs staffinfo . . . 454
1.25.3.1 StaffInfo.pbo . . . 454
1.26 Tip Functions . . . 455
1.26.1 Setup tip functions . . . 456
1.26.2 External tip page . . . 456
1.26.3 pbs article-tiplink . . . 458
1.26.3.1 ArtTips.pbo . . . 459
1.26.3.2 ArtTipsText.pbo . . . 459
1.26.3.3 ArtTipsTextHtml.pbo . . . 460
1.26.3.4 ClassTipsText.pbo . . . 460
1.26.3.5 ClassTipsTextHTML.pbo . . . 460
1.26.3.6 GuideTipsText.pbo . . . 460
1.26.3.7 GuideTipsTextHTML.pbo . . . 460
1.26.3.8 Tips.pbo . . . 461
1.26.3.9 TipsFailed.pbo . . . 461
1.26.3.10 TipsText.pbo . . . 461
1.27 Newsletter Composer Tags . . . 462

1.27.1 Web based management . . . 463
1.27.2 Main template . . . 463
1.27.3 NewsletterAd.pbo . . . 464
1.27.4 NewsletterArticle.pbo . . . 464
1.27.5 Newsletter email . . . 464
1.27.6 Newsletter ini . . . 465
1.28 Omniture . . . 466
1.28.1 Setup Omniture . . . 467
1.28.1.1 Procedure to Implement Omniture . . . 467
1.28.1.1.1 Test Environment . . . 467
1.28.1.1.2 Implementing Omniture . . . 470
1.28.2 pbs omniturecode . . . 471
1.28.2.1 Omniture.pbo . . . 471
1.28.2.2 URL action . . . 473
1.29 US Weather . . . 473
1.29.1 pbs accuwforecast . . . 474
1.29.1.1 AccuWForecast.pbo . . . 475
1.29.2 pbs accuwcurrent . . . 475
1.29.2.1 AccuWCurrent.pbo . . . 476
1.29.3 pbs accuwname . . . 477
1.29.4 pbs accuwmap . . . 477
1.29.4.1 AccuWMap.pbo . . . 478
1.29.5 pbs accuwoptions . . . 478
1.30 European Weather . . . 479
1.30.1 pbs weatherforecast . . . 480
1.30.1.1 WeatherForecast.pbo . . . 480
1.30.1.2 WeatherForecastColItem.pbo . . . 480
1.30.1.3 WeatherForecastItem.pbo . . . 480
1.30.1.4 WeatherForecastRowItem.pbo . . . 481
1.30.2 pbs weathergraph . . . 481
1.30.2.1 WeatherGraph.pbo . . . 481
1.30.3 pbs weatherlastupdate . . . 481
1.30.4 pbs weatherobservations . . . 482
1.30.4.1 WeatherObservation.pbo . . . 482
1.30.4.2 WeatherObservationItem.pbo . . . 482
1.30.5 pbs weatherpicture . . . 482
1.30.5.1 WeatherPicture.pbo . . . 483
1.30.6 pbs weathertext . . . 483
1.30.7 pbs weathertidesunmoon . . . 483
1.30.7.1 WeatherTideSunMoon.pbo . . . 483
1.30.8 Storm weather service . . . 484
1.31 Horoscopes . . . 484
1.31.1 pbs horoscope . . . 485
1.32 TV Guide . . . 485
1.32.1 pbs tv_getoptions . . . 486
1.32.2 pbs tv_LineUpData . . . 487
1.32.2.1 TVLineUpSelect.pbo . . . 487
1.32.2.2 TVLineUpZipSelect.pbo . . . 487
1.32.3 pbs tv_programinfo . . . 487
1.32.3.1 TvPrograminfo.pbo . . . 488
1.32.4 pbs tv_schedule . . . 488
1.32.4.1 TVSchedule.pbo . . . 488

1.32.4.2 TVScheduleHeaderRow.pbo . . . 489
1.32.4.3 TVScheduleRow.pbo . . . 489
1.32.4.4 TVScheduleRowCell.pbo . . . 489
1.33 Movie Guide . . . 489
1.33.1 Link to own reviews . . . 490
1.33.2 pbs mv_movieinfo . . . 490
1.33.2.1 MVMovieInfo.pbo . . . 491
1.33.2.2 MVMovieInfoImageItem.pbo . . . 492
1.33.2.3 MVMovieInfoImageRow.pbo . . . 492
1.33.2.4 MVMovieInfoReviewItem.pbo . . . 492
1.33.3 pbs mv_schedule . . . 492
1.33.3.1 MVSchedule.pbo . . . 493
1.33.3.2 MVScheduleChainFooter.pbo . . . 493
1.33.3.3 MVScheduleChainHeader.pbo . . . 493
1.33.3.4 MVScheduleMovieItem.pbo . . . 493
1.33.3.5 MVScheduleTheaterFooter.pbo . . . 493
1.33.3.6 MVScheduleTheaterHeader.pbo . . . 493
1.33.4 Search form for movie guide . . . 494
1.34 Other Tags . . . 495
1.34.1 pbs calendar . . . 496
1.34.1.1 Calendar.pbo . . . 497
1.34.1.2 CalendarDateItem.pbo . . . 498
1.34.1.3 CalendarWeekItem.pbo . . . 498
1.34.2 pbs Captcha . . . 498
1.34.2.1 Captcha.pbo . . . 499
1.34.3 pbs category . . . 500
1.34.4 pbs categorydescription . . . 500
1.34.5 pbs categoryid . . . 501
1.34.6 pbs catselect . . . 501
1.34.7 pbs datecalc . . . 501
1.34.8 pbs datechoice . . . 503
1.34.9 pbs delimiter . . . 503
1.34.9.1 Delimiter.pbo . . . 503
1.34.10 pbs dingbats . . . 504
1.34.10.1 DingBats.pbo . . . 504
1.34.11 pbs frontpagearchive . . . 504
1.34.11.1 FPArchive.pbo . . . 505
1.34.11.2 FPArchiveItem.pbo . . . 505
1.34.12 pbs getcookie . . . 505
1.34.13 pbs getheaderinfo . . . 505
1.34.13.1 HeaderInfo.pbo . . . 505
1.34.14 pbs GetKeywords . . . 508
1.34.14.1 KeywordList.pbo . . . 509
1.34.15 pbs lastupdated . . . 509
1.34.16 pbs link . . . 510
1.34.17 pbs metarefresh . . . 510
1.34.18 pbs parvariables name="site" . . . 510
1.34.19 pbs PbcsRequestRandomA . . . 510
1.34.20 pbs PbcsRequestRandomB . . . 511
1.34.21 pbs profileid . . . 511
1.34.22 pbs profilename . . . 511
1.34.22.1 ProfileName.pbo . . . 511

1.34.23 pbs prog . . . 511
1.34.24 pbs query . . . 512
1.34.24.1 CustomQuery_queryname.pbo . . . 513
1.34.24.2 CustomQuery_querynameitem.pbo . . . 513
1.34.25 pbs queryparam . . . 513
1.34.26 pbs random . . . 513
1.34.27 pbs script . . . 514
1.34.28 pbs sessioninfo . . . 516
1.34.29 pbs setheaderinfo . . . 517
1.34.30 pbs showdate . . . 518
1.34.31 pbs statuslog . . . 518
1.34.32 pbs taxonomyinfo . . . 518
1.34.32.1 TaxonomyInfo.pbo . . . 519
1.34.32.2 TaxonomyInfoItem.pbo . . . 519
1.34.32.3 TaxonomyInfoItemPathItem.pbo . . . 519
1.34.33 pbs UniquePageID . . . 519
1.34.34 Pop-up calendar . . . 520
1.34.35 sessioninfo with static cache files . . . 520
1.35 Community Site . . . 521
1.35.1 Community site setup . . . 522
1.35.2 Other tags and community site . . . 522
1.35.3 pbs article and community site . . . 522
1.35.4 pbs buildmenu . . . 522
1.35.4.1 MenuGraphicItem.pbo . . . 522
1.35.4.2 MenuTextItem.pbo . . . 523
1.35.5 pbs lightid . . . 523
1.35.6 pbs links on community site . . . 523
1.35.7 pbs sectionfront on community site . . . 523
1.35.8 Community site templates . . . 523
1.36 Xsendmail . . . 523
1.36.1 XSendMail.pbo . . . 524
1.36.2 XSendMail Form . . . 524
1.36.3 XSendMailHTML.pbo . . . 528
1.36.4 XSendMailHTMLItem.pbo . . . 528
1.36.5 XSendMailItem.pbo . . . 528
1.36.6 XSendMail setup . . . 528
1.37 Go.dll . . . 530
1.37.1 guide_GuideAdItem . . . 531
1.37.2 guide_GuideMainItem . . . 531
1.38 Deprecated . . . 531
1.38.1 Forum Tags . . . 532
1.38.1.1 Message filtering . . . 532
1.38.1.2 Message form . . . 533
1.38.1.2.1 ForumNewMessage.pbo . . . 533
1.38.1.2.2 ForumNewMessageExtLogin.pbo . . . 536
1.38.1.2.3 ForumNewMessageUnavailable.pbo . . . 536
1.38.1.3 pbs forumoptions . . . 536
1.38.1.3.1 ForumPreferences.pbo . . . 536
1.38.1.3.2 ForumPreferencesSaved.pbo . . . 538
1.38.1.3.3 ForumPrivateEmail.pbo . . . 538
1.38.1.3.4 ForumPrivateEmailMessage.pbo . . . 540
1.38.1.3.5 ForumPrivateEmailSent.pbo . . . 540

1.38.1.4 pbs news_content 541
1.38.1.4.1 ForumMessage.pbo 542
1.38.1.4.2 ForumMessageAttachmentItem.pbo 544
1.38.1.4.3 ForumMessageThreadItem.pbo 544
1.38.1.4.4 ForumThemeItem.pbo 546
1.38.1.4.5 ForumThemes.pbo 546
1.38.1.4.6 ForumThemesGroupItem.pbo 546
1.38.1.5 pbs news_count 547
1.38.1.6 pbs news_grouplink 547
1.38.1.6.1 ForumGroupLink.pbo 547
1.38.1.7 pbs news_groups 547
1.38.1.7.1 ForumGroupHeadlineItem.pbo 548
1.38.1.7.2 ForumGroupItem.pbo 548
1.38.1.7.3 ForumGroups.pbo 549
1.38.1.8 pbs news_hotthreads 549
1.38.1.8.1 ForumHotThreadItem.pbo 551
1.38.1.8.2 ForumHotThreads.pbo 551
1.38.1.9 pbs news_latest 551
1.38.1.9.1 ForumLatestMessageItem.pbo 553
1.38.1.9.2 ForumLatestMessages.pbo 554
1.38.1.10 pbs news_messages 554
1.38.1.10.1 ForumMessageItem.pbo 554
1.38.1.10.2 ForumMessages.pbo 556
1.38.1.10.3 ForumMessagesMessageThreadItem.pbo 557
1.38.1.10.4 ForumNavBar.pbo 560
1.38.1.10.5 ForumNavBarItem.pbo 560
1.38.1.11 pbs search 560
1.38.1.12 pbs searchresult and forums 562
1.38.1.12.1 SearchResultForum.pbo 562
1.38.1.12.2 SearchResultForumItem.pbo 562
1.38.1.13 pbs userverify 563
1.38.1.13.1 UserVerificationMail.pbo 563
1.38.1.13.2 UserVerify.pbo 563

Tags Manual

Welcome to the SAXOTECH Online Tags Manual. This manual covers the technical aspects of setting up a publication with SAXOTECH Online. It is targeted toward people who make and manage SAXOTECH Online templates. This manual is to be used as a reference manual in conjunction with the Template Training documentation.

PDF Version

Tag Index

Object File Index

Tag Parsing Order

HTML and Includes

Templates

Object Files

Menu Tags

Section Front Tags

Article Tags

Overview Tags

Images

Newsletter Tags

Searching

Custom Content Engine

User Generated Content Tags

User Comments

Q&A Tags

Banner Ads

Links

Page Designer Tags

Net Guest Tags

Subscriptions

User Registration

Staff Listing

Tip Functions

Newsletter Composer Tags

Omniture

US Weather

PDF version

A PDF version of this manual is is available here for download: SAXOTECH Online 7.0.0 Tags Manual.

Search Tags

To perform a search in only the Tags Manual (not the rest of the wiki) use this form:

European Weather

Horoscopes

TV Guide

Movie Guide

Other Tags

Community Site

Xsendmail

Go.dll

Deprecated

PDF Version

A PDF version of the Tags Manual is attached to this page and can be downloaded by clicking the link below.

File	Modified
SAXOTECH Online Tags Manual.pdf	Dec 29, 2009 by Eliane Nysom Pohl

Tag Index

Search Tags

To perform a search in only the Tags Manual (not the rest of the wiki) use this form:

Tag Index

- pbs accuwcurrent
- pbs accuwforecast
- pbs accuwmap
- pbs accuwname
- pbs accuwoptions
- pbs ad
- pbs adregusertargets
- pbs agent
- pbs agentcategory
- pbs article
- pbs article-links
- pbs article-tiplink
- pbs articledate
- pbs articlefield
- pbs articlemultimedia
- pbs artikkel-urls
- pbs buildmenu
- pbs calendar
- pbs Captcha
- pbs category
- pbs categorydescription
- pbs categoryid
- pbs categorytree
- pbs catselect
- pbs cce
- pbs ccedetails
- pbs ccefields
- pbs changepassword
- pbs contentlist
- pbs datecalc
- pbs datechoice
- pbs delimiter
- pbs departmentlist
- pbs departmentoverview
- pbs dingbats
- pbs extra_income
- pbs filecheck

- pbs forumoptions
- pbs frontpagearchive
- pbs ft
- pbs getcontent
- pbs getcookie
- pbs getheaderinfo
- pbs GetKeywords
- pbs global
- pbs history
- pbs horoscope
- pbs imagesearchresult
- pbs include
- pbs keywordindex
- pbs keywordlinks
- pbs LastUpdate for CCE
- pbs lastupdated
- pbs leftmenu
- pbs lightid
- pbs link
- pbs links
- pbs linkscategorylist
- pbs macro
- pbs menu-image
- pbs menu-text
- pbs metarefresh
- pbs mostread
- pbs mv_movieinfo
- pbs mv_schedule
- pbs netguest
- pbs netguest_list
- pbs netguest_questionform
- pbs news_content
- pbs news_count
- pbs news_grouplink
- pbs news_groups
- pbs news_hotthreads
- pbs news_latest
- pbs news_messages
- pbs Newslad
- pbs newsletter_subscriptionform
- pbs newslist
- pbs omniturecode
- pbs oversiktnotiser
- pbs parvariables name="site"
- pbs PbcsRequestRandomA
- pbs PbcsRequestRandomB
- pbs photogallery

- pbs photogallery-latest
- pbs photogallery-links
- pbs photogallery-thumbnails
- pbs pollarchive
- pbs polldata
- pbs pollmini
- pbs pollquestion
- pbs profileid
- pbs profilename
- pbs profiletree
- pbs prog
- pbs ptimages
- pbs ptshowimage
- pbs query
- pbs queryparam
- pbs quest_multipage
- pbs questform
- pbs qwiknavsearch
- pbs random
- pbs reguserinfo
- pbs related
- pbs script
- pbs search
- pbs search_criteria
- pbs search_hits
- pbs search_next_link
- pbs search_previous_link
- pbs search_totalhits
- pbs searchresult
- pbs searchresult and forums
- pbs searchresult without object files
- pbs sectionfront
- pbs sessioninfo
- pbs setheaderinfo
- pbs showdate
- pbs staffinfo
- pbs statuslog
- pbs subcategories
- pbs subscriberdata
- pbs themelinks
- pbs tv_getoptions
- pbs tv_LineUpData
- pbs tv_programinfo
- pbs tv_schedule
- pbs UniquePageID
- pbs userarticleform
- pbs UserComments

- pbs UserCommentsSubmit
- pbs userid
- pbs usersubmitform for UGC
- pbs userverify
- pbs weatherforecast
- pbs weathergraph
- pbs weatherlastupdate
- pbs weatherobservations
- pbs weatherpicture
- pbs weathertext
- pbs weathertidesunmoon
- Pop-up calendar
- Search form

Object File Index

Search Tags

To perform a search in only the Tags Manual (not the rest of the wiki) use this form:

Object File Index

- AccuWCurrent.pbo
- AccuWForecast.pbo
- AccuWMap.pbo
- AdSpot.pbo
- Agent.pbo
- AgentItem.pbo
- Article.pbo
- ArticleAccessControls.pbo
- ArticleAccessControlsItem.pbo
- ArticleByline.pbo
- ArticleConnectionItem.pbo
- ArticleConnectionItem.pbo and images
- ArticleConnections.pbo
- ArticleConnections.pbo and images
- ArticleExtLogin.pbo
- ArticleExtraPictureItem.pbo
- ArticleExtraPictures.pbo
- ArticleFacts.pbo
- ArticleForm.pbo
- ArticleFormMediaItem.pbo
- ArticleFormSectionItem.pbo
- ArticleForumConnectionItem.pbo
- ArticleForumConnections.pbo
- ArticleGalleryConnectionItem.pbo
- ArticleGalleryConnections.pbo
- ArticleGeoDataItem.pbo
- ArticleGeoDataItems.pbo
- ArticleHead.pbo
- ArticleLinkItem.pbo
- ArticleLinks.pbo
- ArticleLocation.pbo
- ArticleModified.pbo
- ArticleMultiMedia.pbo
- ArticleMultiMediaExtension.pbo
- ArticleMultimediaLinkItem.pbo
- ArticleMultiMediaList.pbo
- ArticleMultiMediaListItem.pbo

- ArticleMultimediaListItemExtension.pbo
- ArticleMultimediaListItemXXX.pbo
- ArticleMultiMediaPlaylistItem.pbo
- ArticleParagraph.pbo
- ArticlePhotoGalleryThumbnailItem.pbo
- ArticlePhotoGalleryThumbnails.pbo
- ArticlePicture.pbo
- ArticlePublished.pbo
- ArticleSummary.pbo
- ArticleTaxonomy.pbo
- ArticleTaxonomyItem.pbo
- ArticleTaxonomyItemPathItem.pbo
- ArtTips.pbo
- ArtTipsText.pbo
- ArtTipsTextHtml.pbo
- Calendar.pbo
- CalendarDateItem.pbo
- CalendarWeekItem.pbo
- Captcha.pbo
- CategoryTree.pbo
- CategoryTreeItemActive.pbo
- CategoryTreeItemPassive.pbo
- CategoryTreeItemSeparatorColl.pbo
- CategoryTreeItemSeparatorExp.pbo
- CategoryTreeItemSeparatorNone.pbo
- CCE_ModuleShortName.pbo
- CCE_ModuleShortNameDetails.pbo
- CCE_ModuleShortNameDetailsAccessControls.pbo
- CCE_ModuleShortNameDetailsAccessControlsItem.pbo
- CCE_ModuleShortNameFieldsFieldName.pbo
- CCE_ModuleShortNameFieldsFieldNameItem.pbo
- CCE_ModuleShortNameFieldsFieldNameRow.pbo
- CCE_ModuleShortNameGroupHeader.pbo
- CCE_ModuleShortNameItem.pbo
- CCE_ModuleShortNameItemAccessControls.pbo
- CCE_ModuleShortNameItemAccessControlsItem.pbo
- CCE_ModuleShortNameNavigator.pbo
- CCE_ModuleShortNameNavigatorItem.pbo
- CCE_ModuleShortNamePageLink
- CCE_ModuleShortNamePageLinkItem
- CCE_ModuleShortNamePrintDetail.pbo
- CCE_ModuleShortNameRelated.pbo
- CCE_MPLACE.pbo
- CCE_MPLACEitem.pbo
- CCEExport.pbo
- CCEExportCustomFieldItem.pbo
- CCEExportDateMapItem.pbo

- CCEExportGroupBy.pbo
- CCEExportItem.pbo
- ClassTipsText.pbo
- ClassTipsTextHTML.pbo
- ContentList.pbo
- ContentListByline.pbo
- ContentListItem.pbo
- ContentListItemAccessControls.pbo
- ContentListItemAccessControlsItem.pbo
- ContentListItemTaxonomy.pbo
- ContentListItemTaxonomyItem.pbo
- ContentListItemTaxonomyItemPathItem.pbo
- CustomQuery_queryname.pbo
- CustomQuery_querynameitem.pbo
- Delimiter.pbo
- DepList.pbo
- DepListItem.pbo
- DepOverview.pbo
- DepOverviewGroupHeader.pbo
- DepOverviewHeader.pbo
- DepOverviewItem.pbo
- DingBats.pbo
- Election2008.pbo
- ExpPhotoAlbum.pbo
- ExpPhotoAlbumItem.pbo
- ExpPhotoAlbums.pbo
- FlashBanner.pbo
- ForumGroupHeadlineItem.pbo
- ForumGroupItem.pbo
- ForumGroupLink.pbo
- ForumGroups.pbo
- ForumHotThreadItem.pbo
- ForumHotThreads.pbo
- ForumLatestMessageItem.pbo
- ForumLatestMessages.pbo
- ForumMessage.pbo
- ForumMessageAttachmentItem.pbo
- ForumMessageItem.pbo
- ForumMessages.pbo
- ForumMessagesMessageThreadItem.pbo
- ForumMessageThreadItem.pbo
- ForumNavBar.pbo
- ForumNavBarItem.pbo
- ForumNewMessage.pbo
- ForumNewMessageExtLogin.pbo
- ForumNewMessageUnavailable.pbo
- ForumPreferences.pbo

ForumPreferencesSaved.pbo
ForumPrivateEmail.pbo
ForumPrivateEmailMessage.pbo
ForumPrivateEmailSent.pbo
ForumThemeItem.pbo
ForumThemes.pbo
ForumThemesGroupItem.pbo
FPArchive.pbo
FPArchiveItem.pbo
GuideTipsText.pbo
GuideTipsTextHTML.pbo
HeaderInfo.pbo
History.pbo
HistoryItem.pbo
ImageItem.pbo
ImageItemRowEnd.pbo
ImageItemRowStart.pbo
ImageSearchNavigator.pbo
ImageSearchNavigatorItem.pbo
InvalidArticle.pbo
KeywordIndexHeader.pbo
KeywordIndexLink.pbo
KeywordIndexTitle.pbo
KeywordLinkItem.pbo
KeywordLinks.pbo
KeywordList.pbo
LayerAdSpot.pbo
Links vote form
Links.pbo
LinksItem.pbo
LinksSubcat.pbo
MenuGraphicItem.pbo
MenuTextItem.pbo
MostRead.pbo
MostReadItem.pbo
MostReadItemAccessControls.pbo
MostReadItemAccessControlsItem.pbo
MostReadItemTaxonomy.pbo
MostReadItemTaxonomyItem.pbo
MostReadItemTaxonomyItemPathItem.pbo
MultiPageQuestIndex.pbo
MultiPageQuestPage.pbo
MultiPageQuestPageX.pbo
MultiPageQuestSubmit.pbo
MultiPageQuestWrongAnswer.pbo
MultiQuestAltItem.pbo

- MVMovieInfo.pbo
- MVMovieInfoImageItem.pbo
- MVMovieInfoImageRow.pbo
- MVMovieInfoReviewItem.pbo
- MVSchedule.pbo
- MVScheduleChainFooter.pbo
- MVScheduleChainHeader.pbo
- MVScheduleMovieItem.pbo
- MVScheduleTheaterFooter.pbo
- MVScheduleTheaterHeader.pbo
- NetGuest.pbo
- NetGuestList.pbo
- NetGuestListItem.pbo
- NetGuestQuestionForm.pbo
- NetGuestQuestionItem.pbo
- News.pbo
- NewsItem.pbo
- NewsItemAccessControls.pbo
- NewsItemAccessControlsItem.pbo
- NewsItemArticleConnectionItem.pbo
- NewsItemByline.pbo
- NewsItemConnectionItem.pbo
- NewsItemExternalLinkItem.pbo
- NewsItemExternalLinks.pbo
- NewsItemForumConnectionItem.pbo
- NewsItemForumGroupLink.pbo
- NewsItemGalleryConnectionItem.pbo
- NewsItemInternalLinks.pbo
- NewsItemKeywordLinkItem.pbo
- NewsItemKeywordLinks.pbo
- NewsItemMultiMediaExtension.pbo
- NewsItemParagraphItem.pbo
- NewsItemThemeLinkItem.pbo
- NewsItemThemeLinks.pbo
- NewsletterActivated.pbo
- NewsLetterActivatedError.pbo
- NewsletterAd.pbo
- NewsletterArticle.pbo
- NewsletterOAMSAdHTML.pbo
- NewsletterRemoved.pbo
- NewsLetterRemoveError.pbo
- NewsLetterRemoveMail.pbo
- NewsLetterSubscribeMail.pbo
- NewsLetterSubscriptionForm.pbo
- OamsAdSpot.pbo
- Omniture.pbo
- PhotoGallery.pbo

- PhotoGalleryAccessControls.pbo
- PhotoGalleryAccessControlsItem.pbo
- PhotoGalleryExtLogin.pbo
- PhotoGalleryItemExtraFieldItem.pbo
- PhotoGalleryItemExtraFields.pbo
- PhotoGalleryItemTaxonomyItem.pbo
- PhotoGalleryLatest.pbo
- PhotoGalleryLatestItem.pbo
- PhotoGalleryLatestItemAccessControls.pbo
- PhotoGalleryLatestItemAccessControlsItem.pbo
- PhotoGalleryLatestItemExtraFieldItem.pbo
- PhotoGalleryLatestItemExtraFields.pbo
- PhotoGalleryLatestItemTaxonomy.pbo
- PhotoGalleryLatestItemTaxonomyItem.pbo
- PhotoGalleryThumbnailItem.pbo
- PhotoGalleryThumbnails.pbo
- PhotoGalleryThumbnailsAccessControls.pbo
- PhotoGalleryThumbnailsAccessControlsItem.pbo
- Poll_Error.pbo with pbs quest_multipage
- PollArchiveItem.pbo
- PollDataResults.pbo
- PollDataResultsItem.pbo
- PollDataResultsItemEntry.pbo
- PollForm.pbo
- PollFormAlternatives.pbo
- PollFormAlternativesItem.pbo
- PollFormItem.pbo
- Pollmini.pbo
- PollMiniChart.pbo
- PollMiniNoPoll.pbo
- Polls_Error.pbo with pbs polldata
- Polls_Error.pbo with pbs questform
- Polls_Saved.pbo with pbs polldata
- Polls_Saved.pbo with pbs quest_multipage
- Polls_Saved.pbo with pbs questform
- ProfileName.pbo
- ProfileTree.pbo
- ProfileTreeItem.pbo
- ProfileTreeItemActive.pbo
- ProfileTreeItemPassive.pbo
- ProfileTreeItemSeparatorColl.pbo
- ProfileTreeItemSeparatorExp.pbo
- PwdSent.pbo
- QuestForm.pbo
- QuestFormAlternativesItem.pbo
- QuestFormItem.pbo
- QwikNavSearchForm.pbo

- RelStoriesItem.pbo
- RelStoriesItems.pbo
- SearchError.pbo
- SearchNavigator.pbo
- SearchNavigatorItem.pbo
- SearchResult.pbo
- SearchResultForum.pbo
- SearchResultForumItem.pbo
- SearchResultImage.pbo
- SearchResultImageItem.pbo
- SearchResultImageItemAccessControls.pbo
- SearchResultImageItemAccessControlsItem.pbo
- SearchResultImageRow.pbo
- SearchResultItem.pbo
- SearchResultItemAccessControls.pbo
- SearchResultItemAccessControlsItem.pbo
- SearchResultItemTaxonomy.pbo
- SearchResultItemTaxonomyItem.pbo
- SearchResultItemTaxonomyItemPathItem.pbo
- ShowImage.pbo
- StaffInfo.pbo
- StorySearchNavigator.pbo
- StorySearchNavigatorItem.pbo
- SubcatSimpleLink.pbo
- SubcatSimpleLink1.pbo
- SubscriberData.pbo
- TaxonomyInfo.pbo
- TaxonomyInfoItem.pbo
- TaxonomyInfoItemPathItem.pbo
- ThemeLinkItem.pbo
- ThemeLinkItem.pbo and overview tags
- ThemeLinks.pbo
- ThemeLinks.pbo and pbs related
- Tips.pbo
- Tips.pbo and articles
- TipsText.pbo
- TVLineUpSelect.pbo
- TVLineUpZipSelect.pbo
- TvPrograminfo.pbo
- TVSchedule.pbo
- TVScheduleHeaderRow.pbo
- TVScheduleRow.pbo
- TVScheduleRowCell.pbo
- URL Parameters and overview tags
- UserComments.pbo
- UserCommentsItem.pbo

- UserCommentsSubmit.pbo
- UserCommentsSubmitError.pbo
- Userreginfo.pbo
- UserSubArticles_ArticleFormNotLoggedIn
- UserSubmitForm.pbo
- UserSubmitFormArticle.pbo
- UserSubmitFormArticleCategoryList.pbo
- UserSubmitFormArticleCategoryListItem.pbo
- UserSubmitFormArticleErrorFailed.pbo
- UserSubmitFormArticleErrorInvalidInput.pbo
- UserSubmitFormArticleErrorNotCreator.pbo
- UserSubmitFormArticleErrorNotLoggedIn.pbo
- UserSubmitFormArticleErrorNoUGCAccess.pbo
- UserSubmitFormArticleExtraFields.pbo
- UserSubmitFormArticleExtraFieldsItem.pbo
- UserSubmitFormArticleMediaItem.pbo
- UserSubmitFormArticleSectionItem.pbo
- UserSubmitFormArticleTaxonomyList.pbo
- UserSubmitFormArticleTaxonomyListItem.pbo
- UserSubmitFormCategoryList.pbo
- UserSubmitFormCategoryListItem.pbo
- UserSubmitFormCCE.pbo
- UserSubmitFormCCECategoryList.pbo
- UserSubmitFormCCECategoryListItem.pbo
- UserSubmitFormCCEErrorFailed.pbo
- UserSubmitFormCCEErrorInvalidInput.pbo
- UserSubmitFormCCEErrorNotCreator.pbo
- UserSubmitFormCCEErrorNotLoggedIn.pbo
- UserSubmitFormCCEErrorNoUGCAccess.pbo
- UserSubmitFormCCEFieldList.pbo
- UserSubmitFormCCEFieldListItem.pbo
- UserSubmitFormCCEModule.pbo
- UserSubmitFormCCERecurrencePattern.pbo
- UserSubmitFormCCETaxonomyList.pbo
- UserSubmitFormCCETaxonomyListItem.pbo
- UserSubmitFormErrorFailed.pbo
- UserSubmitFormErrorInvalidInput.pbo
- UserSubmitFormErrorNotCreator.pbo
- UserSubmitFormErrorNotLoggedIn.pbo
- UserSubmitFormErrorNoUGCAccess.pbo
- UserSubmitFormExtraFields.pbo
- UserSubmitFormExtraFieldsItem.pbo
- UserSubmitFormPhotoGallery.pbo
- UserSubmitFormPhotoGalleryCategoryList.pbo
- UserSubmitFormPhotoGalleryCategoryListItem.pbo
- UserSubmitFormPhotoGalleryErrorFailed.pbo
- UserSubmitFormPhotoGalleryErrorInvalidInput.pbo

- UserSubmitFormPhotoGalleryErrorNotCreator.pbo
- UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo
- UserSubmitFormPhotoGalleryErrorNoUGCAccess.pbo
- UserSubmitFormPhotoGalleryExtraFields.pbo
- UserSubmitFormPhotoGalleryExtraFieldsItem.pbo
- UserSubmitFormPhotoGalleryPhotoItem.pbo
- UserSubmitFormPhotoGalleryTaxonomyList.pbo
- UserSubmitFormPhotoGalleryTaxonomyListItem.pbo
- UserSubmitFormTaxonomyList.pbo
- UserSubmitFormTaxonomyListItem.pbo
- UserVerificationMail.pbo
- UserVerify.pbo
- WeatherForecast.pbo
- WeatherForecastColItem.pbo
- WeatherForecastItem.pbo
- WeatherForecastRowItem.pbo
- WeatherGraph.pbo
- WeatherObservation.pbo
- WeatherObservationItem.pbo
- WeatherPicture.pbo
- WeatherTideSunMoon.pbo
- XSendMail.pbo
- XSendMailHTML.pbo
- XSendMailHTMLItem.pbo
- XSendMailItem.pbo

Tag Parsing Order

The following is an incomplete list of pbs tags and the order in which they are parsed.

Tags at the top of the list are parsed first. Portions of this list will need further analysis to determine whether any additional tags are processed at given points. These points are marked with "**additional tags may be processed here**".

Also note that some tags are referenced by only part of their name. For example, when looking for <pbs:classifieds> tags, the pbcs.dll only looks for "class".

ProcDiverseTags

pbs:subscribercat/pbs:abokat

pbs:subscriberdata/pbs:abondata

pbs:sessioninfo/pbs:#sessioninfo#

pbs:subscribermcactive/pbs:abonhkaktiv

pbs:redirecturl

pbs:changepassword/pbs:byttpassord

pbs:queryparam

pbs:getkeywords

pbs:include

pbs:getcontent

pbs:getcookie

pbs:customtag

pbs:random

pbs:shop

additional tags may be processed here

pbs:datecalc

pbs:datechoice/pbs:datovalg

pbs:options

pbs:filecheck

pbs:fontface

pbs:frontpagearchive/pbs:forsidearchive/pbs:pdfarchive

pbs:categorymovie/pbs:kategorimovie

pbs:buildmenu

pbs:historikk

pbs:horoscope

pbs:login

pbs:kategorioptions

pbs:logininfo

pbs:lm/pbs:ft

pbs:categoryid/pbs:kategoriid

additional tags may be processed here

pbs:lightid

pbs:newuserform

pbs:ntb

pbs:ntbkategori

pbs:agentcategory

pbs:ntbsitat

pbs:mostread

pbs:profilename

pbs:profileid

pbs:paymentdata

pbs:regcompany

pbs:reguser

pbs:searchresult

pbs:imagesearchresult

pbs:showpicturespecial

pbs:target

pbs:temalinker/pbs:stikkordlinker/pbs:keywordlinker

additional tags may be processed here

pbs:reguserinfo

pbs:userid

pbs:usersearch

pbs:subscriberuserid/pbs:abouserid

pbs:userregform

pbs:lastupdated/pbs:sistoppdatert/pbs:visartikkeldato/pbs:visdato

pbs:valutakurs

pbs:valutaopts

pbs:prog

pbs:rediraction

pbs:timestamp

pbs:catselect

pbs:categorytree/pbs:kategoritree

pbs:profiletree

pbs:photogallery-latest

pbs:photogallery

pbs:photogallery-link

pbs:photogallery-thumbnails

pbs:articlemultimedia

pbs:usersubmitform/pbs:userarticleform/pbs:userartikkelform

pbs:metarefresh

pbs:qwiknavsearch

pbs:calendar

pbs:multimedialist

pbs:xmlget

pbs:pdfindexfiles

pbs:extuserinfo

pbs:validatetag

pbs:setheaderinfo

pbs:getheaderinfo

pbs:captcha

ProcInnholdTags

pbs:search

pbs:delimiter

pbs:macro

pbs:query

pbs:script

pbs:dingbats

additional tags may be processed here

pbs:linksoverview/pbs:linkeroverview

pbs:linkscategorylist/pbs:linkerkategorilist

pbs:linksvoteform/pbs:linkervoteform

pbs:linksvoteresult/pbs:linkervoteresult

pbs:linksreglinkform/pbs:linkerreglinkform

pbs:linksmaincategoryid/pbs:linkermainkategoriid

pbs:linkssubcategoryid/pbs:linkersubkategoriid

pbs:linksmaincategory/pbs:linkermainkategori

pbs:linkssubcategory/pbs:linkersubkategori

pbs:linkssubregion/pbs:linkersubregion

pbs:link

pbs:misc

tlDiverse

pbs:agent

pbs:searchres

pbs:imagesearch

pbs:

tlPersonalia

pbs:pers

pbs:staff

pbs:avdeling

pbs:department

tlForside

pbs:frontpage/pbs:forside

pbs:advforside

tlArtikkel

pbs:weather

pb:article/pbs:artikkel

tlPortal

pbs:accuw

pbs:pt

pbs:userverify

pbs:yp

pbs:news

pbs:forum

pbs:tt

pbs:creg

pbs:result

tlOversikt

pbs:oversikt

pbs:utgivelseoversikt

pbs:theme/pbs:temaoversikt

pbs:keywordindex/pbs:stikkordindeks

tlAgenda

pbs:ds

pbs:cinema

pbs:event

tlPolls

pbs:poll

pbs:quest

tlRuban

pbs:class

pbs:rub

pbs:advertising

tlInnhold

pbs:misc

pbs:link

pbs:macro

pbs:script

pbs:search

pbs:delimiter

pbs:dingbats

pbs:query

tlAds

pbs:ad

tlVariables

pbs:#

tlNoCache

pbs:!

tlNetGuest

pbs:netguest

tlTVGuide

pbs:tv

pbs:mv

tlNewsList

pbs:newslist/pbs:sistenytt

tlNabolaget

pbs:nabolaget

tlCCE

pbs:cce

tlOmniture

pbs:omniturecode

HTML and Includes

Include tags and options are used to include content not only from your site but also from other sites. This section also includes information about the HTML document structure and meta tags.

- Document structure
- Fonts
- Meta tags
  - Article90.pbo
  - ArticleHead90.pbo
  - ArticleHead.pbo meta tag
- pbs filecheck
- pbs getcontent
- pbs include
- Server side includes
- Special graphics directory

Document structure

Framework

There are certain tags that should be present in your document. These are shown in the example below.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3 <head>
4 <title>SAXOTECH Online Times</title>
5 <meta name="Description" content="Saxotech SAXOTECH Online Demosite">
6 <meta name="Keywords" content="Demosite, SAXOTECH Online">
7 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
8 <link href="/includes/Style.css" rel="stylesheet" type="text/css"
title="Stylesheet">
9 <link href="/includes/TV.css" rel="stylesheet" type="text/css" title="TV Styles">
10 <script language="JavaScript1.2" src="/includes/functions.js"
type="text/javascript"> </script>
11 <script language="JavaScript" src="/apps/cookiesearchparams.js"
type="text/javascript"> </script>
12 </head>
13 <body bgcolor="#FFFFFF" text="#000000" link="#003366" vlink="#003366"
alink="#003366">
14 Your webpage goes here.
15 </body>
16 </html>
```

Tags

1	The document type is set up. This is used to define which HTML tags the browser should expect. If this tag is not present, the document is treated as HTML version 2.0.
2	The HTML tag starts. This tag must enclose the entire document. In the example, the tag is closed on line 16.
3	We start the HTML header. The header contains information about the document and links to resources placed outside the document.
4	Contains the title for the document.
5	Sets the description of the document.
6	Sets keywords for the document.
7	Contains information about the character set used on the page.

8	Links to the main style sheet. By linking to the style sheet, it will be loaded only once. This makes pages load faster.
9	Loads the style sheet for TV/Movies.
10	Contains a link to the shared JavaScript functions used on the website.
11	Contains a link to the cookie management library.
12	Closes the head of the document.
13	Starts the body of the document. Here the background color and the color for links are set.
14	Includes the text that is the document.
15	Closes the body tag.
16	Closes the HTML tag. It should always be the last line in the document.

Fonts

One of the biggest obstacles when trying to keep a uniform layout on a website is to control the use of fonts.

Style sheets

Another way to define fonts is to use style sheets. When you use style sheets, you define how the different items should look in the beginning of the document. An introduction to style sheets can be found at http://www.w3.org/MarkUp/Guide/Style.html. An example of a simple style sheet is below:

```
<style type="text/css">
	A:link,A:visited {
		text-decoration:none;
		color:#566983
	}
	A.example:link,A:visited {
		text-decoration: underline;
		color:#FF0000
	}
	.categoryname {
		font-family: arial,Helvetica;
		font-size: 10px;
		color: #3399cc;
	}
	.title {
		font-family: arial;
		font-size: 16px;
		color: #3E4A5A;
		font-weight : bold;
	}
	#title2{
		font-family: Times;
		font-size: 16px;
		color: #3E8A5A;
		font-weight : bold;
	}
	H1{
		font-family: Times;
		font-size: 16px;
	}
	h2 {
		font-family: Arial, Helvetica;
		font-size: 20px;
		text-decoration:none;
		font-weight : bolder;
		margin : 0;
	}
	P{
	margin-bottom: 0px; margin-top: 0px;
	}
</style>
```

A	Defines the basic look of the anchor tag. The anchor tag is used for links.
A.example	Used when you want to have another look for the link. When you want to use this version, put class="example" inside the A tag.
category name	Used on the category name. Enclose the text with a div tag like `<div class=categoryname>Cateogy</div>`

H1	Used by the tag `<h1>....</h1>` H1 stands for Heading one, and is usually output with a large font size. By retooling the basic HTML codes with styles, older browsers will display the website better. You have the same function for the H2 tag.
P	P stands for paragraph and can be used to enclose paragraphs of text.
title	This style works the same way as the `*categoryname*` style, but it is used for the title.
#title2	This is an example of an ID style. ID can be used instead of class. If you want to know more about the differences between class and ID, please see www.W3C.org.

Publicus.ini
You can define the fonts you want to use in the Publicus.ini. However, this method of managing fonts was created before style sheets were available and should no longer be used. You define the font faces the following way:

```
Fontface1=Helvetica,Arial,Sans-Serif
Fontface2=Palatino,Times,Serif
```

You can define two font faces to be used throughout the site. When you define fonts, you should always remember to define multiple fonts. You should also include the generic font name for users who do not have the font you specified installed on their system.

Here are the generic names and the fonts to which they are usually translated:

```
Sans-serif = Helvetica,Arial,Geneva
Serif = Times, Garamond, Palatino
Monospace = Courier, Terminal
Cursive = Brush Script, Comic
```

In the templates
When you would like to use the font defined in the ini file, you use the following font tag:

```
<font face="<pbs:fontface1>">Text text text</font>
```

SAXOTECH Online will replace the pbs tag with the font information from the ini file.

In the object files
You can also use the defined fonts in object files. You specify the font tags the following way:

```
<font face="<%fontface1%>">Text tex text</font>
```

SAXOTECH Online will replace the tag with the font specified in the ini file.

Font size

The traditional way of controlling font size is by using

```
<font size=#>
```

where # is a number between 1 and 6. In the SAXOTECH Online tags that do not support object files, or where you do not use object files, you can usually set font sizes in the tag.

Colors
In HTML colors are set by:

```
COLOR = #rrggbb
```

The color attribute sets the text color.

```
#rrggbb
```

is a hexadecimal (base 16) red-green-blue triplet used to specify the RGB color.

To set the background color of a web page, use the following

```
<body>
```

tag:

```
<body bgcolor="#rrggbb"> Rest of document goes here </body>
```

Meta tags

Meta tags are set in the HTML Head tag. One of the advantages of setting the HTML Head is that the story will be indexed by search engines.

You need to use the **<pbs:article>** tag twice, once to extract information for the HTML Head and once for the HTML Body.

The pbs / pbo file description you find below does not show all the variables available in each file. These files are only for illustration.

gen_art.pbs

```
<!--the following include will take care of the "HTML head" part of the document.-->
<!--#include virtual="/includes/art_head.inc"-->
<!--the actual article will be shown using the following pbs tag-->
*<pbs:article>*
```

art_head.inc

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<!--this tag is actually the same the one in gen_art.pbs but will use a different
Objectclass thus a different set of pbo files-->
*<pbs:article objectclass=90>*
</head>
```

- Article90.pbo

- ArticleHead90.pbo
- ArticleHead.pbo meta tag

Article90.pbo

```
<!--You must remove all HTML tags in the Head and Summary values-->
<title><%Head%></title>
<meta name="Description" content="<%Summary$t%>">
```

ArticleHead90.pbo

```
<!--the Head variable in Article90.pbo uses this file-->
<%Heading$t%>
```

ArticleHead.pbo meta tag

```
<!--it seems like some crawlers are looking for <h1> tags in the body of a document-->
[%<h1><%Heading%></h1>%]
[%<%Heading2%>%]
```

pbs filecheck

<pbs:filecheck>

You can also include images based on categories. In this way, if an image does not exist, you have a fallback image. If you use this tag too many places, your site will be very slow.

Example

```
<img src="<pbs:filecheck filename="/images/logo_#category#.gif"
fallback="/images/logo_default.gif">">
```

Options

#Category#	This variable will be replaced with the category ID of the category you are viewing.
Fallback	This field specifies the fallback image if there is no image for the category you are viewing.
Filename	The name of the file that should be included.

#MainCategory#	This variable will be replaced with the category ID of the main category for the category you are viewing.
	Placing
	`*pboscript*`
	in the tag enables you to use pboscript functions in the filename. Example:
	`<pbs:filecheck pboscript filename="/graphics/logo[%{ActiveZone}-french%].jpg">`

#SectionCate#	This variable will be replaced with the category ID of the sectionfrontÕs category and not the category for the story.
Pboscript	

Offsite image include

This function allows you to include a file from another website and use image processing functions on the image. This function is often used when including images from web cameras.

The tag looks like this:

```
<img src="/apps/pbcsi.dll/urlget?url=http://webcam/big.jpg&MaxW=120&Q=100&noborder" border="0">
```

Options

Image options	The function supports all the image options. For more information, please see Image Settings.
URL	Sets the URL of the image you would like to download.

Other uses

You can also use this function to process local images. The tag can look like this:

```
<img src="/apps/pbcsi.dll/urlget?url=/graphics/cl_<pbs:categoryid>.jpg&MaxW=120&Q=100&noborder" border="0" alt="" align="right">
```

pbs getcontent

<pbs:getcontent>

This function allows you to include a file from another website. If the operation fails, the system will leave a comment about how long it took and the URL that could not be downloaded.

Example

```
<pbs:!getcontent interval=15 cacheid="CLICKTV"
url="http://www.clicktv.com/index.asp?cid=AM1">
```

Options

Cacheid	This is the local file name, which should be used for caching. The default setting is the **URL** option. This option is optional.
Cachetime	Controls how long the pbs:getcontent tag is cached in seconds. The default setting is -1, indicating that it does not use any cache. This option is optional.

Imgsrc	Adds a string before x in
	`src="x"`
	of the URL response content. This is used to change local, relative paths in the HTML of the external site to point to the external site. Default is blank. This option is optional.An example is to change
	`src="/apps/g/image.jpg"`
	to
	`scr=http://www.dailynews.com/apps/g/image.jpg by using imgsrc="http://www.dailynews.com."`

Interval	This option is the same as **Cachetime**. The default is -1 (do not cache). This option is optional.
StatusOutput	If set to 1, the status comment normally added on errors is added on success as well. The default setting is 0. This option is optional.
Timeout	Timeout of the HTTP request in seconds. The default setting is 5. This is optional.
URL	This is the URL that should be pulled from the other site. This option is mandatory.
ConnectTimeout	Limit the time it is allowed for the client to wait for the server to return the 1st byte of the response. Mainly made in order to avoid foreign web servers to take forever before providing answer. Default is 5,000 ms. Available from 6.3.0sr11 and up.
UseProxy	[0|1] default 1. This setting can be used in hosted environments where all out-bound communication is done through proxies. These proxies can cache result return from external source. By setting this parameter to 0 the system will not use out-bound proxies
Username & Password	Its also possible to send Basic Authentication with the tag <pbs:getcontent url="http://username:password@saxonews.com/apps/pbcs.dll/frontpage">

pbs include

<pbs:include>

A more advanced version allows you to put a category-specific include in a general file. This include type is slower than the standard include, and is performed later in the template assembly. This tag should only be used when there is no other way to achieve the result.

```
<pbs:!include virtual="/includes/header.inc" cachetime="10">
```

Options

Artupdate	Used instead of cache time. With this option, the file will be updated when changes are made to stories in your database.
CacheQualifiers	Please see Cachequalifiers for more information about caching.
Cachetime	Sets the time the include should be cached before it is refreshed. This option is only used when the tag is running uncached.
Categorymatch	The file will only be included if you are viewing the category specified in categorymatch.
Interval	A synonym for cache time.
Nocache	Disables caching for the file. Usually, the system generates a cache file that is used by all similar includes. When you enable this option, the system caches the content in the main cache file. This option is used like this: `nocache=1`
Useaid	This function includes the date, category and story number in the cache file name. The function can be useful for caching story forum messages (on the story page). The function is used like this: `Useaid=1`
Useprofile	This option is only necessary if you are using profiles and the template changes based on profile. With this option in the tag, a cache file will be created for each profile.
Virtual	Used for specifying the file you would like to include.
skiptemplatecache	Allows the user to skip template caching. ⚠ This option should not be used unless SAXOTECH has been informed.
ShowError	Prevents the exception message from pbs include tag to appear on the page. Default is 1. From version 7.0.0sr11 it also hides template read errors.

Server side includes

Server side includes is a function that allows you to split up your HTML code into multiple files. By doing this, you will only have to maintain code that is used multiple places in one file. This ensures that all pages that should look the same way do look the same way.

Standard includes

SAXOTECH Online supports the standard way of including files. A typical file that includes the standard header and footer would look something like this:

```
<!--#include virtual="/includefiles/header.pbs"-->
Some text that is unique.
<!--#include virtual="/includefiles/footer.pbs"-->
```

The included files contain the information that is present on all pages.

Including files based on category

In addition to including files that are to be used on all categories, it is also possible to have different files included based on the category being viewed. If you use the code:

```
<!--#include virtual="/includes/poll_#category#.inc"-->
```

`#Category#` is replaced by the category ID of the category being viewed. This is the same name that is used for menu links and so on.

If no specific file exists for the category, the system will look for a file with the category name replaced by default. If this file does not exist, SAXOTECH Online will not display anything.

If you are on a subsection, you can use the following piece of code to include a file based on the category ID of the main category:

```
<!--#include virtual="/includes/poll_#maincategory#.inc"-->
```

If no file exists for the main category, the system will fall back to using a file named

```
*default*
```

When template mappings are used

When you map one category to another, the system will first look for the original category, and then the category to which it is mapped. For more information, please see Template mapping#opinion_arts.

For example:

You have created a **gen_art.pbs**, which will be used on all stories. For the categories *Opinion* and *Arts* you would like to include a file in the template. You put the code:

```
<!--#include virtual="inc_#category#.incÓ-->
```

in your file and create **inc_arts.inc**. Instead of creating another copy of the file, which is to be used when the category is *Opinion,* you set the map **opinion=arts** in Publicus.ini instead.

When the system is displaying a story in the *Opinion* category, the system will first look for inc_opinion.inc. If this file does not exist, it will look for inc_arts.inc. If you use this function, you avoid duplicating your code in multiple files.

When profiles are used

In addition to including files that are to be used on all profiles, it is also possible to have different files included based on the profile being viewed. If you use the code:

```
<!--#include virtual="/includes/poll_#profile#.inc"-->
#Profile#
```

is replaced by the profile ID of the profile being viewed. This is the same name that is used for menu links and so on.

If no specific file exists for the profile, the system will look for a file with the profile name replaced by "default." If this file does not exist, SAXOTECH Online will not display anything.

When sectioncat is used

When you use profiles, stories can be placed in another category than the one through which they are displayed. To include files based on the category in which the stories are displayed, use the variable

```
 #sectioncat#
```

It is used like this:

```
<!-- #include virtual="/includes/#sectioncat#_header.inc"-->
```

Including files based on zone

When you use zones, the system will automatically try to include a file for the zone you are in.

The regular include:

```
<!--#include virtual="/includes/zone.inc"-->
```

will look for the file /includes/zone.inc when zones are not used. When zones are used, the system will first try to include the file **/includes/$zoneid/zone.inc**.

```
*$zoneid*
```

is the ID you have given to the zone. This ID is usually a three letter abbreviation of the zone name. You must create a folder under the includes folder with the same name as the zone ID. If the folder does not exist, the system will fall back to regular behavior.

Including files based on week day

This option allows you to include different files for different days. The option is used like this:

```
<!--#include virtual="/includes/header_#dow#.inc"-->
*#dow#*
```

is a number between 1 and 7. Sunday is 1, Monday is 2, etc. The default will be used if no file matching the day is found.

Special graphics directory

The special graphics directory is placed outside the base folder for the website. The default location for this folder is **\inetpub\g**. This folder is

mounted as a virtual non-browsable, readable folder in the website configuration as **/g**. The folder contains graphic files that are used by SAXOTECH Online and is not editable by the website operator.

The most important files are:

b.gif	Black
t.gif	Transparent
w.gif	White

These files are 1 by 1 pixel and are used for drawing lines and spacing.

In addition, this folder contains other graphics files that are used by the system. You can use these files in your design. These files will always be present on the server.

Templates

The use of templates is what makes SAXOTECH Online a dynamic publishing solution. The website design is based on the templates. The templates contain codes that pull information from the database, allowing the user to display and control information using tags rather than programming.

Template placement
By default, all templates are placed in the folder **/mal** on the web server. Enabling the option TemplatePath in the Publicus.ini, allows you to use the folder **/templates** instead.

- Export templates
  - article_xml_asset.pbo
  - article_xml_connection.pbo
  - article_xml_paragraph.pbo
  - article_xml.pbo
    - article_xml_accesscontrolsitem.pbo
  - templates_Sms_xml
- Folder structure
- Setting MIME Types
- Special templates
- Template mapping
- Template names and URLs
- Zones

Export templates

Export templates can be created to customize the XML output of certain items. Once customized, they must be placed in \objects\export to be used by the system. Customization of these templates will not be overwritten during system updates.

The following export templates are available:

- article_xml_asset.pbo
- article_xml_connection.pbo
- article_xml_paragraph.pbo
- article_xml.pbo
  - article_xml_accesscontrolsitem.pbo
- templates_Sms_xml

Sample code from each template is included in the relevant sections. Unless otherwise noted, only the fields in the template are available.

article_xml_asset.pbo

The following is an example of the article_xml_asset.pbo:

```
<asset>
  <Filename><%Filename%></Filename>
</asset>
```

article_xml_connection.pbo

The following is an example of the article_xml_connection.pbo:

```
<articlekey>
<date><%Date%></date>
<category><%Category%></category>
<articlenumber><%ArtNo%></articlenumber>
<reference><%Reference%></reference>
</articlekey>
```

article_xml_paragraph.pbo

The following is an example of the article_xml_paragraph.pbo:

```
<paragraph id="<%Block%>" html="<%IsHTML%>">
[%
<heading><%Heading%></heading>
%]
<body><%Body%></body>
[%{FactsBody}
<facts align="<%FactsAlign%>">
[%
<heading><%FactsHeading%></heading>
%]
<body><%FactsBody%></body>
</facts>
%][%
<image align="<%ImgAlign%>" mimetype="<%ImgMimeType%>" width="<%ImgWidth%>"
height="<%ImgHeight%>">
<filename><%ImgFilename%></filename>
%][%
<caption-leadin><%Caption-Leadin%></caption-leadin>
%][%
<caption><%Caption%></caption>
%][%
<photo><%ImgPhoto%></photo>
%][%{ImgFilename}
</image>
%]
</paragraph>
```

article_xml.pbo

<%Site%>	Contains the site code.
<%Date%>	Contains the stories date, in format.YYYYMMDD
<%Category%>	Contains the category ID.
<%ArtNo%>	Contains the story number.
<%URL%>	Contains the link to the story.
<%SuperHeading%>	Contains the value of the kicker.
<%Heading%>	Contains the Heading.
<%Heading2%>	Contains the sub heading.
<%NetHeading%>	Contains the net heading.
<%Location%>	Used to output the location data..
<%Summary%>	Outputs the summary for the story.
<%Byline%>	Outputs the byline.
<%Taxonomy%>	Outputs the taxonomy words attached to the story.
<%Theme%>	Contains the story theme.
<%Keywords%>	Contains the keywords attached to the story.
<%FactsHeading%>	Outputs the title for the fact box.
<%FactsBody%>	Outputs the text for the fact box.

<%Connections%>	Outputs the links for connected items.
<%Paragraphs%>	Outputs all of the paragraphs for the story.
<%SEOLabels%>	Outputs the SEO label of the story.
<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of article_xml_accesscontrolsitem.pbo

In the article_xml.pbo it is possible to use custom fields as variables in the format by using the following:

```
<%FIELDS_FieldName%>
```

The following is an example of the article_xml.pbo:

```
<?xml version="1.0" encoding="[%<%CharSet%>%|%iso-8859-1%]"?>
<!DOCTYPE article SYSTEM "http://www.publicus.com/xml/pbcs5_article.dtd">
<article sitecode="<%Site%>">
<properties priority="<%Priority%>" subarticle="<%Subarticle%>" status="<%Status%>"
note="<%Note%>">
<date><%Date%></date>
[%{PublishFrom|PublishTo}
<publishtime from="<%PublishFrom%>" to="<%PublishTo%>"/>
%]
<category><%Category%></category>
<articlenumber><%ArtNo%></articlenumber>
<reference><%Reference%></reference>
[%
<subclasstype frontpage="<%FPSubClass%>" section="<%SecSubClass%>"
article="<%ArtSubClass%>"/>
%][%
<forum><%Forum%></forum>
%]
<rating><%Rating%></rating>
<theme><%Theme%></theme>
<keywords><%Keywords%></keywords>
[%
<taxonomy><%Taxonomy%></taxonomy>
%]
<created time="<%Created%>" by="<%CreatedBy%>"/>
<modified time="<%Modified%>" by="<%ModifiedBy%>"/>
[%
<page><%FIELDS_PAGE%></page>
%][%
<section><%FIELDS_SECTION%></section>
%][%
<source><%Source%></source>
%]
</properties>
[%{channels}
<channels WEB="<%ChWEB%>" WAP="<%ChWAP%>" PDA="<%ChPDA%>" TXTTV="<%ChTXTTV%>"
SMS="<%ChSMS%>"/>
%]
<url><%URL%></url>
[%
<connections>
<%Connections%>
```

```
</connections>
%]
<head>
[%
<superheading><%SuperHeading%></superheading>
%]
<heading><%Heading%></heading>
[%
<heading2><%Heading2%></heading2>
%][%
<netheading><%NetHeading%></netheading>
%][%
<summary><%Summary%></summary>
%][%
<sectionsummary><%SectionSummary%></sectionsummary>
%][%
<byline><%Byline%></byline>
%][%
<location><%Location%></location>
%][%
<image mimetype="<%ImgMimeType%>" width="<%ImgWidth%>" height="<%ImgHeight%>">
<filename><%ImgFilename%></filename>
%][%
<caption-leadin><%Caption-Leadin%></caption-leadin>
%][%
<caption><%Caption%></caption>
%][%
<photo><%ImgPhoto%></photo>
%][%{ImgFilename}
</image>
%]
</head>
[%
<facts align="<%FactsAlign%>">
<heading><%FactsHeading%></heading>
<body><%FactsBody%></body>
</facts>
%][%<assets><%assets%></assets>
```

```
%][%<%Paragraphs%>
%]</article>
```

article_xml_accesscontrolsitem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

templates_Sms_xml

sms_xml.pbo

The following is an example of the sms_xml.pbo:

```
<?xml version="1.0" encoding="[%<%CharSet%>%|%iso-8859-1%]"?>
```

```
<!DOCTYPE smsarticle SYSTEM "http://www.publicus.com/xml/pbcs5_smsarticle.dtd">
```

```
<smsarticle sitecode="<%Site%>">
```

```
<properties priority="<%Priority%>" status="<%Status%>">
```

```
<date><%Date%></date>
```

```
<category><%Category%></category>
```

```
<articlenumber><%ArtNo%></articlenumber>
```

```
<created time="<%Created%>" by="<%CreatedBy%>"/>
```

```
<modified time="<%Modified%>" by="<%ModifiedBy%>"/>
```

```
</properties>
```

```
<url><%URL%></url>
```

```
<head>
```

```
<heading><%Heading%></heading>
```

```
[%
```

```
<byline><%Byline%></byline>
```

```
%][%
```

```
<location><%Location%></location>
```

```
%]
```

```
</head>
```

```
<body>
```

```
<content><%Body%></content>
```

```
</body>
```

```
</smsarticle>
```

Folder structure

Your website should have the following folder structure:

/	The root folder for the website. A file called default.htm must be placed in this folder. It must redirect to the front page URL.
/ads	This folder is used for storing ads graphics.
/adlogs	This folder is used for logging by the ad system.
/.cache	This folder is used by the SAXOTECH Online cache engine.
/includes	Files that are included by templates go here.
/light	This folder is used for Community Site sites.
/misc/misc	Files that should be parsed by SAXOTECH Online, but are not templates should be placed in this folder.
/templates	The main folder for template files.
/templates/macros	The main folder for macros.
/templates/macros/<category>	Macros for one category.
/templates/objects	The main folder for object files.
/templates/objects/<category>	Object files that are available for a single category.
/templates/objects/sectioncat/<sectioncat>	
/templates/objects/profiles/<profileID>	
/templates/objects/template/<templatename>	

Setting MIME Types

MIME types are used for identifying content to the browser or application you use for downloading information. By default, these are predefined in the web server based on the file extension. When you download a JPG file, the server automatically sends the MIME type image/jpeg.

You can override MIME type directly in the URL. This is useful if you would like to have your web page treated as XML instead of HTML. As long as you are dealing with text, you only have to specify the last part of the MIME type. Add

```
&mime=xml
```

to set the document's MIME type to text/XML. This allows you to set a different MIME type for special templates. For example, pages that are going to be read by an application, like for an RSS feed.

If you would like to set the MIME type to something different, you can also specify the first part of the string like this:

```
&mime=image%2fgif
```

or

```
&mime=x-application%2fspacemonkey
```

"%2f" is the code for "/." The "/" must be written as

```
%2f
```

Special templates

The following special templates also exist.

Error template

This template is shown when a requested object cannot be found on the server. The file is called **404.pbs** and must be placed in the template folder. In order for this to work, the web server must be set up to pass error requests for the error type 404 to SAXOTECH Online. This must also be done if you are using the Alias function.

This page should not contain any SAXOTECH Online tags. The page is shown for every missing object you have, and it can generate a lot of extra traffic if you have tags on this page.

Refresh page

If you have made changes to your error document and want to refresh them without waiting for the cache to expire, you can use this URL:

```
http://www.yoursite.com/apps/pbcs.dll/error?nocache=1
```

Robots.txt

This file is not a SAXOTECH Online file. It is a standard web server file. The file controls which parts of your website robots will be allowed to access. This file should be present on your website. It must be placed in the root folder of your website.

Template mapping

In the Publicus.ini, there is a **TEMPLATEMAP** section that contains information about which templates are to be used. Use this section to define mappings so that you do not have to make multiple copies of the template files. For example:

```
[TEMPLATEMAP]
_FART=_ART
_FUART=_ART
_UART=_ART
_OVRN=_OVR
opinion=arts
ABOUT_FART=ADVERTISE_ART
ABOUT_OVR=ADVERTISE_OVR
ADMAIL_ART=STATIC_ART
PROFILE1007_ART=BUSINESS_ART
```

All of these options must be placed under the **TEMPLATEMAP** section of the ini file. The different types of templates are described below.

Mapping types

about_fart=advertise_art	With this option you map a specific story template to another category.
about_ovr=advertise_ovr	With this option you map a specific sectionfront to another category.
_fart=_art	This option maps all front page stories to the normal story template. A front page story is a story that has been linked to from a story of the type front page.
_fuart=_art	This option maps all front page subarticle stories to the normal story template. A front page subarticle is a story with the *Subarticle *checkbox marked in story properties, and which is linked to from a story of the type front page.
opinion=arts	This option maps all templates from the *Opinion* category to the *Arts* category, for all template types. When you use include files with the category as part of the name, the system will look for both categories. For more information, please see Server side includes .
_ovrn=_ovr	This option maps all section fronts with notices to the standard overview template. A notice is a story with the notice checkbox marked in the story properties. This template type is obsolete.
_uart=_art	This option maps all subarticles to the story template. Subarticles are stories with the **Subarticle** checkbox marked in story properties. Example:For a sports subarticle, the system will first look for sports_uart.pbs. If this file does not exist, the system will look for sports_art.pbs. If this template does not exist, the system will look for gen_uart.pbs. The last template the system will look for is gen_art.pbs. If this template does not exist, the system will fail.
profile1007_art=business_art	This option maps all articles with profile 1007 to the business article template. The profile is required in the story url for the mapping to work, such as /article/20140615/ARTICLE/140619999/1007

AlwaysCheckMainCatMap=1

This option introduces a rule that makes the system check main category mappings even if sub-category mappings exist, as long as the resulting file from the sub-category match did not exist. This means that if you have mapped *News* to *Sports,* News01 will use the Sports template if the mappning exist and the resulting file did not exist.

AlwaysCheckMainCatMap=0

[TEMPLATEMAP]
sport_ovr=saxo02_ovr
sport02_ovr=golf_ovr

golf_ovr.pbs do not exsit

/section/sport02 --> gen_ovr.pbs (Will not check main category)

Saxotech Online will look for a file called golf_ovr.pbs if that file not exist, Online will pick our fallback file gen_ovr.pbs

AlwaysCheckMainCatMap=1

[TEMPLATEMAP]
sport_ovr=saxo02_ovr
sport02_ovr=golf_ovr

golf_ovr.pbs do not exsit

/section/sport02 --> saxo02_ovr.pbs (Will now check main category)

Saxotech Online will look for a file called golf_ovr.pbs if that file not exist, Online will instead of using our fallback file go to the MainCat template check. Beware that the mapning must be in the TemplateMap section before this AlwaysCheckMainCatMap setting will work.

AlwaysCheckSubCatMap=1

As default Saxotech Online will go to the MainCat from the DetailCat instead of the SubCat. Change that by adding AlwaysCheckSubCatMap=1

/section/john <-- main category
/section/john01 <-- subcategory
/section/john0101 <-- detailcategory

Other Methods

If the differences between the templates are more subtle, like color changes and different parts that are to be included, you can use SAXOTECH Online tags to include different information on different categories. These tags are described in Other Tags.

Template names and URLs

All templates in SAXOTECH Online must be named according to a predefined structure. This naming structure is based on the template type and category. All template files have the .pbs extension. All template files must be placed in the template folder under the root directory for the website.

Front page template

The front page template is called **frontpage.pbs.**
For Danish, Swedish and Norwegien sites the default is **forside.pbs**.
This is the template that is used by the website's homepage.
The front page is called by accessing the following URL:

```
/apps/pbcs.dll/frontpage
```

You must place a default document in the root folder of the web server that redirects to this address.

Template URL parameter

From version 6.8.0sr14, the frontage action supports the 'template=templateName* URL parameter.
The template lookup will follow this order for sites that is not using Danish, Norwegian or Swedish language.:

```
Frontpage_[templateName].pbs (6.8.0sr14)
gen_[templateName].pbs (6.8.0sr14)
Frontpage.pbs
gen.pbs
Forside.pbs
```

For Danish, Swedish and Norwegian sites, the order is:

```
Forside_[templateName].pbs (6.8.0sr14)
gen_[templateName].pbs (6.8.0sr14)
Forside.pbs
gen.pbs
Frontpage.pbs
```

Sectionfront templates

Sectionfronts are accessed through the URL:

```
/apps/pbcs.dll/section?Category=XXX.
```

Replace

```
XXX
```

with the name of the category.

Default template

The default sectionfront template is called **gen_ovr.pbs**. This template is used if no template exists for the category, or no mappings exist in the ini file.
Section templates are named using the following structure:

- The first part of the name signifies the category.
- **Gen** is the general category and is used when no other template file exists.
- The second part of the name indicates the template type. For example, **ovr** signifies that the template is an overview/section template.

Single category template

If you want to use a special template for the category News, the file should be named news_ovr.pbs. The gen portion of the name has been replaced with the category ID. For subcategories of news that are named news01....news99, the template named news_ovr.pbs will be used. This is a built-in default mapping in the system.

Sectionfront with notices

If you use the SAXOTECH Online function **Notices** on a category, the **gen_ovrn.pbs** template is used instead of gen_ovr.pbs.
Notices is a function that allows you to filter out short stories from the main section overview. Whether or not a category uses notices is defined in the category setup. To use a category template for a section where you use notices, the template should be named

```
<category>_ovrn.pbs.
```

Template URL parameter

From version 6.8.0sr14, the way the section action handles the template URL parameter have changed slightly.
The template lookup will follow this order:

```
[category]_[templateName]_ovr.pbs (6.8.0sr14)
gen_[templateName]_ovr.pbs (6.8.0sr14)
[category]_[templateName].pbs
gen_[templateName].pbs
```

The _ovr part of the template may change in accordance to the other parameters, so it can also be _ovrn.pbs.

Story templates

Default story template
Story templates are named with a structure similar to sectionfronts. The default template for stories is called **gen_art.pbs**.
To use a template for only one category, name the template:

```
<category>_art.pbs
```

As with section overviews, stories placed in subcategories named category01..99 will automatically use the same template as the main category,

if no specific template exists.

Substory template

A substory is a story that has the **Substory** type selected in the **Story type** drop-down on the story properties page. Substories are not listed by the Newslist or Sectionfront tags. Substories use the template **gen_uart.pbs**. They are searchable by SAXOTECH Online Search powered by FAST, but are not searchable by the old SAXOTECH SQL server-based search.

Front page linked story

This template is used when you link to a story from the front page and have the option

```
FrontpageFlag=1
```

in the pbs newslist. The template is called **gen_fart.pbs**.

Front page linked subarticle

The **gen_fuart.pbs** is used by all subarticles listed with a main story when the

```
Frontpageflag=1
```

option is used.

Unique templates for the same story (Template URL parameter)

To use a different template for the same article, create the template and then use the template name with the

```
<%link%>&template=template_name
```

option.

For example, the same story might be shown in the sections *Local* and *Sports* and its appearance should correspond to the section in which it is being viewed. To use the template **gen_sports.pbs**, enter

```
<%link%>&template=sports.
```

From version 6.8.0sr14, the way the article action handles the template URL parameter have changed slightly.
The template lookup will follow this order:

```
[category]_[templateName]_art.pbs (6.8.0sr14)
gen_[templateName]_art.pbs (6.8.0sr14)
[category]_[templateName].pbs
gen_[templateName].pbs
```

The _art part of the template may change in accordance to the other parameters. E.g. _uart.pbs, _fart.pbs.

Other templates

Other templates are discussed with their tags.

Zones

Checking zone cookie

You can check if the zone cookie is set by accessing the front page through this URL:

```
/apps/pbcs.dll/frontpage?Checkcookie=PBCSZone
```

This will bring you to the template called **PBCSzone_setcookie.pbs**.

Object Files

Object files are the pieces used for designing the layout of the output. They give you total control of how the information is presented. The object file for a tag is described with the tag.

- About Object Files
- Article classes
- Global function
- Global variables
- Inline objects
- Insert Header and Footer
    - InsertHeader.pbo InsertFooter.pbo
        - InsertByline.pbo
        - InsertBylineItem.pbo
        - InsertCCETypes.pbo
        - InsertCCETypesItem.pbo
        - InsertContextCategory.pbo
        - InsertContextProfile.pbo
        - InsertSessionInfo.pbo
        - InsertTaxonomyList.pbo
        - InsertTaxonomyListItem.pbo
- Localdata.ini
- Object script
- Odd and Even variable picture handling
- pbs macro
    - Election2008.pbo
- Position-based design
- Session function
- System variables

Object files are the pieces that make up a story page, a Classifieds page, or any other dynamically-displayed section of your website. For example, if you were to separate a news story into its various pieces, you might have a headline, a byline, the story text, a footer, and so on. Each of these elements could be considered as an object file. You can arrange, modify and manipulate object files to display the story in different ways. You can make the headline bold and at the top of the story, or you could put it all lowercase at the bottom of the story.

This chapter describes the functionality of object files and scripting. It contains general information about how object files work. The specific information about how an object file works is placed with the tag that uses the object file.

This chapter also contains information about object script, which allows you to build conditional object files. The functionality we call Article classes, which allows the user to select a specific object file for one story, is also described in this chapter.

The use of tags and object files
It is not possible to set a variable in an object file as shown in this example:

```
<%?new_var="pbs:datecalc>"%>
```

If used in this manner, the variable new_var will be equal to <pbs:datecalc> instead of the date desired.

When you want a variable with the results of a pbs tag in it, and not the tag itself, call it as part of a macro and make the variable a global variable. In macros you can use pbs tags like `<pbs:newslist>`, `<pbs:article>,` or `<pbs:macro>` tag (nested macros). An example of this is:

```
<pbs:macro name="twodaysago" test="<pbs:datecalc dateformat=yyyymmdd days=-1>">
```

and then in the macro:

```
<%?@new_var=#test#%>
```

About Object Files

Trigger use of object files

In some tags, you must use the option **useobjects=1** to get the tag to use an object file. Other tags will automatically use the object file if it exists. This feature is documented for each individual tag.

Naming of object files

Object files for most tags are named with the following structure:

```
<Function><Objectclass>.pbo
```

Object files for the newslist tag are named with the following structure:

```
<Function><Objectclass><ArticleClass>_<Position>_<Zone>.pbo
```

Object files for the article tag are named with the following structure:

```
<Function><Objectclass><ArticleClass>_<Zone>.pbo
```

Article class	Article class is selected by the users when they are editing stories. The design options on the story properties page sets the article class for the story. Article class is a letter from A to Z. Article class must be defined in the ini file. For more information, please see [Article classes.
Function	Function is the name of the object file. This name is used by the tag to find the correct object file.
Object class	If you are working with a headline object file, you can have 99 different styles of headlines, numbered 1 to 99. Object class is set by the designer in the tag on the template and has a numerical value from 1 to 99.
Zone	The zone value is used to show different layouts for different geographical zones. The system only checks for a zone version if the non-zoned version is found. For example the file News2a_NOR.pbo will not be used if News2a.pbo does not exist. The zone cookie must also be set.

Directory structure

Object files are placed in a folder under the template folder called **objects**. Inside the **objects** folder, you can create a folder for each category. These folders are named after the category ID of the category, not the category name. By using the folder structure, you do not need to have all object files in the main folder. This makes it easier to keep an overview of the object files.

Folder selection

When you use object files for a tag, SAXOTECH Online looks in the **object** folder for the correct object file to use. The system will look for object files in the following order:

```
/light/<lightid>/objects/
```

This folder is used for Community Site pages. The lightid is the customer ID for the Community Site customer.

```
/objects/themes/<theme>
```

If a story has a theme, the system will look for a folder with the theme's name.

```
/objects/profiles/<profile>
```

If profiles are used, the system will look for a folder with the profile number.

```
/objects/template/<template>
```

If template is defined in the url(example: <domain>/article/20120101/CATEGORY/123456789&template=print), the system will look for a folder with the template name, in this case for the folder "print".

```
/objects/sectioncat/<SectionCat>
```

The system looks for a folder matching the section category. The section category is the category in which the story is shown, and not the category in which the story is placed.

```
/objects/sectioncat/<MainSectionCat>
```

The system looks for a folder matching the section category. If the section category is a subcategory, the main category will be checked as well.

When using category-tree with more than two levels, two digits will be removed at a time. In the following example, five lookups will be performed.

Example 1

```
NEWS010101
NEWS0101
NEWS01
NEWS
default
```

If a site has used an odd number of digits on a subcategory, this will be regarded as the only subcategory. For example:

Example 2
(Note that there are three digits on the subcategory NEWS123)

```
NEWS123
NEWS
default
/objects/<category>
```

The system will look for a folder with the same name as the category ID.

```
/objects/<maincategory>
```

The system will look for a folder with the same name as the main category if the story is placed on a subcategory.

```
/objects/
```

The system will look in the default folder if no other match is found.

```
Global
```

The global folder is the system's fallback folder and is only used when no object files are found. This folder is maintained by SAXOTECH.

Selecting which object files to use

There are 16 possible object files that can be used. The following example illustrates which object file will be used first.

You have a newslist tag on the front page, which uses object class 1. You have multiple stories in the system, some with article classes and some without. The zone is set to ENG. When the page is built, the system will look for object files in the following way:

The newslist tag will start looking for the object file that best corresponds with the specifications on the tag. It will start somewhere in the list below and move downwards, if it does not find the corresponding file.

Newsitem1A_5.pbo	The system will use this object file if the article is in position 5, and has article class A. Position, article class and object class match.
Newsitem1A_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem1A.pbo	The system will use this object file if there is not an object file for the position, but there is one for the article class. Article class and object class match.
Newsitem1A_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem1_5.pbo	This file is selected if there is no object file for the article class, but there is one for the position the article has in the list. Position and object class match.
Newsitem1_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem1.pbo	This file is used if there is no object file for the position or article class. Only object class matches.
Newsitem1_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
NewsitemA_5.pbo	This file is used if there is no match for the object class, but article class and position match.
NewsitemA_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
NewsitemA.pbo	This file is used if there is no match for position or object class. Only article class matches.
NewsitemA_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG
Newsitem_5.pbo	This file is used if there is no match for article class or object class. Only position matches.
Newsitem_5_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.
Newsitem.pbo	This file is used if there us no match for position, article class or object class.
Newsitem_ENG.pbo	The system will use this file if the previous file is found and the zone is set to ENG.

⚠ You can set up object files in subfolders (example /templates/objects/news/NewsItem.pbo). If you are using the Newslist tag on the sectionfront for the category *News,* the system will use NewsItem.pbo in *News* instead of the default NewsItem.pbo. Further, if you are using the Newslist tag on the sectionfront for the category *NEWS04,* the system will use NewsItem.pbo in *News,* instead of using NewsItem1A_2.pbo in the objects folder.

File descriptions

The 16 files are listed below in the order of increasing specification detail of the object file name.

NewsItem.pbo	Default. This file is used if the newslist tag contains no position, article class or object class.
NewsItem_5.pbo	This file is used if the newslist tag has `position=5` When the newslist tag also contains object class or article class, it will start looking for other files. You can also use the **<%Counter%>** variable in NewsItem files to match position.
NewsItemA.pbo	This file is used if the newslist tag has `articleclass=A` When the newslist tag also contains object class or position, it will start looking for other files.

NewsItemA_5.pbo	This file is used if the newslist tag has
	`articleclass=A`
	and
	`position=5`
	When the newslist tag also contains object class, it will start looking for other files. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1.pbo	This file is used if the newslist tag has `objectclass=1` When the newslist tag also contains article class or position, it will start looking for other files.

NewsItem1_5.pbo	This file is selected if the newslist tag has
	`objectclass=1`
	and
	`position=5`
	When the newslist tag also contains article class, it will start looking for other files. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1A.pbo	The system will use this object file if the newslist tag has
	`objectclass=1`
	and
	`articleclass=A`
	When the newslist tag also contains a position, it will start looking for other files.

NewsItem1A_5.pbo	The system will use this object file if the story has
	`objectclass=1`
	,
	`articleclass=A`
	and is in
	`position 5`
	You can also use the **<%Counter%>** variable in NewsItem files to match position.

If zone is set to ENG:	
NewsItem_ENG.pbo	This file is used if the newslist tag contains no position, article class or object class and the zone is set to ENG.
NewsItem_5_ENG.pbo	This file is used if the newslist tag has `position=5` and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.
NewsItemA_ENG.pbo	This file is used if the newslist tag has `articleclass=A` and the zone is set to ENG.

NewsItemA_5_ENG.pbo	This file is used if the newslist tag has
	`articleclass=A`
	`position=5`
	and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1_ENG.pbo	This file is used if the newslist tag has `objectclass=1` and the zone is set to ENG.

NewsItem1_5_ENG.pbo	This file is selected if the newslist tag has
	`objectclass=1`
	`position=5`
	and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.

NewsItem1A_ENG.pbo	The system will use this object file if the newslist tag has
	`objectclass=1`
	`articleclass=A`
	and the zone is set to ENG.

NewsItem1A_5_ENG.pbo	The system will use this object file if the story has
	`objectclass=1`
	`articleclass=A`
	is in
	`position 5`
	and the zone is set to ENG. You can also use the **<%Counter%>** variable in NewsItem files to match position.

Article classes

Article classes allow you to predefine multiple templates that the users can select when they write stories. Before the user can do this, the following must be set up:

Setup
The following code must be present in the Publicus.ini file for the site.

FRONTPAGESUBCLASS section

```
a=Breaking news
b=Major event
```

SECTIONSUBCLASS section

```
a=Breaking news
b=Major event
```

ARTSUBCLASS section

```
a=Breaking news
b=Major event
```

Templates

Article	The article template must be built using object files.
Front page	The front page must be built using the Pbs newslist. This is the only overview tag that supports article classes.
Sectionfront	The sectionfront must be built using the Pbs newslist. This is the only overview tag that supports article classes.

Object files

The naming of the object files for use with article classes is as follows.

Article	For the object files used by the article tag, you add the article class letter to the end of the name. The object file for the heading will be named **ArticleHeadingA.pbo**. For object class 2 and article class B, the file must be named **ArticleHeading2B.pbo**.
Newslist	The Pbs newslist tag must be used on both the front page and the sectionfront.The object files used by the newslist tag are called **News .pbo** and **NewsItem.pbo**. When you use an object class, the object class number is input before the article subclass and the files are named **NewsItem2A.pbo** and **NewsItem2B.pbo**. The object files can also be placed in a sub folder with the same name as the category. The file **News2.pbo** is used for all stories.

Example
A system has been set up with two subclasses for the front page. These are called *Breaking News* and *Major Event*. *Breaking News* is used when something happens during the day, and *Major Event* is used when something major happens, like earthquakes, assassinations, and so on. Stories have been written using all the different subclasses. To present the information in the database, the following tags and object fields are used.

Tag

The following newslist tag is used to build the main page.
<pbs:newslist categories="news%,sports%" count=15 days=2 priosort useobjects=1 Objectclass=1>

Object files

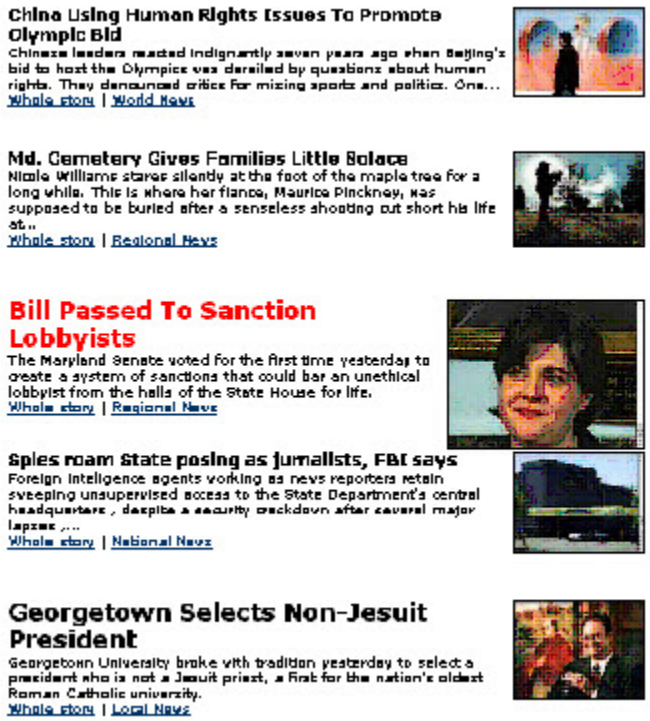
News1.pbo	The main file that is used for the entire list.
	`<font face="<pbs:fontface1>" size=1>`
	`<%items%>`
	`</font>`

NewsItem1.pbo	Used for the regular items. `<br clear=all>` `[%{Imagelink}<img` `src="<%Imagelink%>&MaxW=100"` `width=100 border=0 hspace=1 vspace=1` `align="right">%]` `<font face=Verdana` `size=2><b><%\Title%></b></font> ` `<font face=Verdana` `size=1><%Leadin%></font> ` `<div align="left"><font face=Verdana` `size=1>` `<a href="<%Link%>">Whole story</a> |` `<a` `href="<%pbcs.dll%>/section?Category=<` `%Category%>">` `<%Categoryname%></a></font></div>` ` ` `</font>`
NewsItem1A.pbo	Used for all items marked as *Breaking News.* `<br clear=all>` `[%{Imagelink}<img` `src="<%Imagelink%>&MaxW=100"` `width=100 border=0 hspace=1 vspace=1` `align="right">%]` `<font face=Verdana size=4>` `<b><%\Title%></b></font> ` `<font face=Verdana` `size=1><%Leadin%></font> ` `<div align="left"><font face=Verdana` `size=1>` `<a href="<%Link%>">Whole story</a> |` `<a` `href="<%pbcs.dll%>/section?Category=<` `%Category%>">` `<%Categoryname%></a></font></div>` ` ` `</font>`

NewsItem1B.pbo	Used for all items marked as *Major Event.*

```
<br clear=all>
[%{Imagelink}<img
src="<%Imagelink%>&MaxW=150"
width=150 border=0 hspace=1 vspace=1
align="right">%]
<font face=Verdana size=4
color="#FF0000">
<b><%\Title%></b></font><br>
<font face=Verdana
size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana
size=1>
<a href="<%Link%>">Whole story</a> |
<a
href="<%pbcs.dll%>/section?Category=<
%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

Example page
With A and B class stories:



Global function

The global function is used to output the global and system variables in a template. User defined global variables and system variables can be used. The function uses an inline object like this:

```
<pbs:#global# object="<%currentdate$d(\"dddd, mmm. d\")%>">
```

Global variables

The following conditions apply to global variables:

- A Global variable can be set from any object file or Macro. The created global variable can be accessed from any tag which is processed after the creation of the global variable during the processing of a request.
- If a global variable is used to execute a script which is system intensive then make sure that the tag is cached by either using a macro or using cachetime on the tag if it is supported.
- You create global variables by adding ?@ like this:

You can write global variables using object script. You can use the global variables in the same way as all other variables.

Set the value of ThisIsATest to Hello World)

```
<%?@ThisIsATest="Hello World"%>
```

Set the value of ThisIsATest to The same value of ThisIsATest. You can set the value of a global variable to any other variable available in either global variables or variables available within the tag or Macro

```
<%?@SaveAVave=#@ThisIsATest#%>
```

Testing

You can use the following code to test if global variables are available in the object file:

```
[%{@GlobalVariablesSupported=1}This objectfile supports global variables.%]
```

To set the global variable, use this syntax:

```
<%?@bar="value to be used"%>
```

Examples

Macro1.pbo	Sets the variable @bar: `<%?@bar="12"%>`
Macro2.pbo	Activates the variable @bar: `<%@bar%>`

Call both macros from a PBS-file	`<pbs:name="Macro1">` `<pbs:name="Macro2">`

Pbs:globalvars tag

It is now possible to pull the values of global variables out in the templates using the pbs:globalvars tag.

The advantage of doing it with the globalvars tag, is that this tag is processed before any other tags on the page, so it can be used to set parameters in other tags.

`This will output the value of @SomeVar. If this variable does not have any value, it will output 5`

```
<pbs:globalvars name="@SomeVar" Default="5">
```

This can be used to set cache time like this:

```
<pbs:macro name="SomeMacro" cachtime=<pbs:globalvars name="@SomeVar" Default="5">>
```

Inline objects

Inline objects allow you to format the string output with `<pbs:#variable#>` tags. The object is formatted like this:

```
<pbs:#ureg_message# object="[%<%variable$r(\"Wrong=Incorrect\")%>%]">
```

The quotation marks must be preceded by a backslash for the function to work.

Example

```
<a href="<pbs:!#LastReferrer#
object="[%<%Variable%>%|%javascript:history.back();%]">">
```

Inline objects can also be used as a macro to run, for example, a simple check. If used in this way, it is not necessary to have a macro file in the macros folder.

Example

```
<pbs:macro inlineobject="[%{category}Yes%|%No%]">
```

Insert Header and Footer

This feature is available from versions 7.1.0sr19, 7.2.0sr11.

Insert Header and Footer

It is possible to have the system always render header and a footer object files and embed the result into all request responses.

The feature will make the system render the request as usual and then afterwards, it will:

- Look for </head> and if found render the special InsertHeader.pbo object file and insert it right before the found tag.
- Look for </body> and if found render the special InsertFooter.pbo object file and insert it right before the found tag.

The feature will work for all requests against the pbcs.dll, meaning everything from frontpages, section fronts, article displays, UGC submits, search requests, etc., **so be careful about what is done in those object files**.

Caching

For the display of content items, additional variables are available in the header and footer object files. To make the feature perform those variables are rendered before the outermost caching layer within the application, meaning the fullpage cache for stories and CCE objects, and the action cache for galleries. The context dependent and user specific variables are rerendered on all requests.

Enabling / Disabling

In case the feature is to be used, the following needs to be enabled

```
[GENERAL]
InsertHeaderFooter=1
```

See publicus.ini's GENERAL section for more information on the setting.

InsertHeader.pbo InsertFooter.pbo

The InsertHeader.pbo and InsertFooter.pbo object files support the following variables:

Variables available for all requests

<#ContextCategory#>	Outputs the context category through the object file of InsertContextCategory.pbo.
<#ContextProfile#>	Outputs the context profile through the object file of InsertContextProfile.pbo.
<#SessionInfo#>	Outputs session information through the object file of InsertSessionInfo.pbo that supports the same variables like when rendering a pbs:sessionInfo tag.

Variables available when rendering a content item (story, gallery, CCE object)

Content item specific variables are only available when rendering content item display actions.

When rendering any content item

<%Type%>	Outputs the content item type. Either 'story', 'gallery' or 'cce'.
<%ItemGUID%>	Outputs the GUID of the content item.
<%Title%>	Outputs the title of the content item.
<%TaxonomyList%>	Outputs a list of taxonomywords rendered through the object files of InsertTaxonomyList.pbo and InsertTaxonomyListItem.pbo.

When rendering a story

<%NetTitle%>	Outputs the net title of the story.
<%CategoryID%>	Outputs the ID of the story's category.
<%CategoryName%>	Outputs the display name of the story's category.

<%MainProfileID%>	Outputs the story's main profile ID.
<%MainProfileName%>	Outputs the story's main profile name.
<%LegacyByline%>	Outputs the byline if smart byline is not used.
<%Byline%>	Outputs the smart byline through the object files of InsertByline.pbo and InsertBylineItem.pbo.
<%Keywords%>	Outputs the keywords of the story in a commalist.
<%Preview%> **From version 7.2.0sr17**	Outputs 1 if the story render is actually done as a preview and not through the normal article render.

When rendering a gallery

<%CategoryID%>	Outputs the ID of the gallery's category.
<%CategoryName%>	Outputs the display name of the gallery's category.

When rendering a CCE object

<%ItemID%>	Outputs the objects ID.
<%ModuleID%>	Outputs the ID of the object's module.
<%ModuleName%>	Outputs the display name of the object's module.
<%ModuleType%>	Outputs the type of the object's module.
<%ClassID%>	Outputs the ID of the object's class.
<%ClassName%>	Outputs the display name of the object's class.
<%TypeID%>	Outputs the ID of the object's main type.
<%TypeName%>	Outputs the display name of the object's main type.
<%CCETypes%>	Outputs the types of the object through the object files of InsertCCETypes.pbo and InsertCCETypesItem.pbo.

InsertByline.pbo

The following variables are available in the object file.

<%Items%>	The smart byline staff members, each rendered by the object file of InsertBylineItem.pbo.
<%TotalCount%>	The number of items in the list.

InsertBylineItem.pbo

The following variables are available in the object file.

<%StaffID%>	The staff member's ID.
<%FirstName%>	The staff member's first name.
<%LastName%>	The staff member's last name.
<%Counter%>	The item's number in the list.
<%TotalCount%>	The number of items in the list.

InsertCCETypes.pbo

The following variables are available in the object file.

<%Items%>	The CCE types, each rendered by the object file of InsertCCETypesItem.pbo.
<%TotalCount%>	The number of items in the list.

InsertCCETypesItem.pbo

The following variables are available in the object file.

<%CCEType%>	The CCE type ID.
<%CCETypeName%>	The CCE type's display name.
<%Counter%>	The item's number in the list.
<%TotalCount%>	The number of items in the list.

InsertContextCategory.pbo

The following variables are available in the object file.

<%CategoryID%>	Outputs the ID of the context category.
<%CategoryName%>	Outputs the display name of the context category.

InsertContextProfile.pbo

The following variables are available in the object file.

<%ProfileID%>	Outputs the ID of the context profile.
<%ProfileName%>	Outputs the display name of the context profile.

InsertSessionInfo.pbo

The variables available are the same as the ones available when rendering the <pbs:sessioninfo> tag.

InsertTaxonomyList.pbo

The following variables are available in the object file.

<%Items%>	The taxonomy words, each rendered by the object file of InsertTaxonomyListItem.pbo.
<%TotalCount%>	The number of items in the list.

InsertTaxonomyListItem.pbo

The following variables are available in the object file.

<%ID%>	The taxonomy word's ID.
<%Name%>	The taxonomy word's display name.
<%Counter%>	The item's number in the list.
<%TotalCount%>	The number of items in the list.

Localdata.ini

Localdata.ini is a file used to hold your own permanent global variables. This is useful for storing values that you use in multiple places on your site. The file has the following sections:

Category keys	Key lookup is used to look up values based on variables:
	`[Category_Keys]`
	`Sports=Ball sports`
	`Sports01=Football`
	`Sports(\d)=Sport$1`
	`News=Bar`
	This allows you to perform a lookup on a field and replace the value of the field with the value from the key lookup table. In the example above, a field called *category* with the value *Sports* will show *Ball sports*. If the field has the value sports01, the value returned will also be *Ball sports*. The reason for this is that the field is matched in the order they are listed. On the first match found, the system will stop looking. For this reason you must put the most generic match option at the bottom of the list.
	Options
	`$k`
	means exact match. The match is case insensitive.
	`$K`
	means regular expression. The regular expression follows the standard rules for regular expressions.

STATICVALUES	`[STATICVALUES]`
	`MaxImageWidth=200`
	`NoRecordsFoundMessage=No Customers Found`
	This will allow you to use the variable
	`<%MaxImageWidth%>`
	in all macros.
	localdata.ini settings based on the domain in the URL.
	`[STATICVALUES_helloworld.com]`
	`MaxImageWidth=400`
	If the section exists the variables from that section will be used. if not the system will look for the variables in [STATICVALUES].

Object script

Object script is the functionality that allows you to make conditional decisions in object files based on the content of the fields in the object file. In addition, object script allows you to do arithmetic with the values of numerical variables.

Test if a variable has content

If a piece of code is only to be output if the variable contains information, you can enclose the whole line or section in `[% ... %]` tags.

In the following example, the break after the field will only be output when

```
<%text%>
```

has content:

```
[% <%text%> <br> %]
```

Multiple variables

If you have more than one field in the block, you must write the test like this:

```
[%{Link}<a href="<%Link%>"><%Title%></a>%]
```

With two or more variables, you must specify which variable you want to check. This is done by enclosing the variable in {}.

Example
The following piece of code is only output when the variable Imagelink is not blank:

```
[%{Imagelink}
  <table border="0" cellspacing="0" cellpadding="2" align="right" bgcolor="#EEEEEE">
    <tr><td><img src="<%Imagelink%>&MaxW=200&MaxH=150&Q=85" border=0></td></tr>
    <tr><td align="left"><%Caption%></td></tr>
  </table>
%]
```

Reverse test

You can also use this as a reverse test using ! which is the operator NOT:

```
[%{!Link}There is no link%]
```

In this example, the text will be output when there is no link.

Nesting tests

You can nest tests inside each other. In this example, the table row with the caption will only be output if the caption exists:

```
[%{Imagelink}<table border="0" align="right" bgcolor="#EEEEEE">
<tr><td><img src="<%Imagelink%>&MaxW=200&MaxH=150&Q=85" border=0></td></tr>
[%<tr><td align="left"><%Caption%></td></tr>%]
</table>
%]
```

Alternate output

You can also output different text based on the value of a variable. The format looks like this:

```
[%{variable}The variable has a value %|% The variable has no value %]
```

Example
In this example, the title will be a link only when the link field has a value:

```
[%{Link}<a href="<%Link%>"><%Title%></a>%|%<%Title%>%]
```

Value tests

It also possible to check against the value of a field. This allows you to do a more advanced test than merely checking if a field has a value.

Basic math

You can preform the following tests:

```
{Width>240} which is true when <%Width%> is larger than 240.
  {Width<240} which is true when <%Width%> is smaller than 240.
  {Width=240} which is true when <%Width%> equals 240.
  {Width!240} which is true when <%Width%> does not equal 240.
```

Example
In the following example, the field *Width* is checked to see if it contains a value that is lager than 240. Code is output based on the check:

```
[%{Width<240}
  <img src="/g/t.gif" width="240">
%|%
  <img src="/g/t.gif" width="<%Width%>">
%]
```

The width of the image will be at least 240 pixels. If the width field has a higher value, the image will retain its value.

Odd or even test

This function allows you to build a newslist in which you can alternate pictures on the left or right. For more information, please see Odd and even.
Use the following test to check if the variable is odd:

```
[%{Width is odd}Output this text when width is odd%]
```

Use the following test to check if the variable is even:

```
[%{Width is even}Output this text when width is even%]
```

Advanced value test

You can also compare the numerical value of two variables. This expands on the basic test that compares the variable to a number. When you use this function, the second variable must be enclosed in hash signs #. The test must be placed inside curly brackets {}.

Example 1
In this example, the last changed date is only output if the story has been modified after it was written. You must use the ISO versions of the variables when you define a date test.

```
Published: <%PublishedDate%>
[%{ISOModifiedDate>#ISOPublishedDate#} - Last changed: <%ModifiedDate%> %]
```

Example 2
In this example, the position of the picture is adjusted, depending on whether the picture is higher than it is wide. If the photo is wider than it is high, the photo will be aligned left. If the photo is higher than it is wide, it will be aligned right. The photo will also be scaled to a maximum width of 200 pixels or a maximum height of 300 pixels. This function is built into the image function:

```
<img src="<%ImgLink%>&MaxH=300MaxW=200" Align="[%{Width>#Height#}Left%|%Right%]" />
```

Multiple tests

It is also possible to perform multiple tests to determine if the code should be output. This function uses an ampersand **&** for **AND**. It uses a pipe **|** for **OR**. & has a higher priority than |.

Example

```
[%{Category=INTERNET&ImageLink|Category=NEWS}<font size=4>%|%<font size=2>%]
```

In this example, the font size will be set to 4 if one of the following statements is true:

- The category is *Internet* **and** ImageLink contains a value (not blank).
- The category is *News.*

Formatting dates

To use date formatting, you must use the operator **d**. The options below are available for this operator.

If available, try using ISO versions of date and time variables:

```
<%isodate%>
```

```
<%isodatetime%>
```

and so on. The <%isodate%> of an evergreen article will always be 99999999hhmm, where hhmm is the site's rollover time. If you do not want the hhmm on the date, use **$c(1,8)** instead of date formatting. This will return the IsoDate for most articles and 99999999 for evergreen articles.

$D Converts ISODateTime into Extended format. For example: `<%ISODateTime$D%>` will output in a GMT minus local time offset, like this *`Wed, 13 Feb 2008 16:16:00 -0400`*.

Please check available variables for the object file in question.

am/pm	Outputs the time in 12 hour format.
c	Date in a predefined short version and time in a predefined long version.
d	The number of the day of the month, without leading zero.
dd	The number of the day of the month, with leading zero.
ddd	Short name of the day (Mon, Tue, Wed, etc.).
dddd	Long name of the day (Monday, Tuesday, etc.).
ddddd	Date in a predefined short version.
dddddd	Date in a predefined long version.

h	Hour without leading zero.
hh	Hour with leading zero.
m	Month number without leading zero.
mm	Month number with leading zero. mm can also mean minute if it is preceded by hh.
mmm	Short name of the month (Jan, Feb, Mar, etc.).
mmmm	Long name of the month (January, February, etc.).
n	Minute without leading zero. M is used for month, so N is used instead.
nn	Minute with leading zero.
s	Second without leading zero.
ss	Second with leading zero.
t	Time in a predefined short version.
tt	Time in a predefined long version.
yy	Year with two numbers. For example 07.
yyyy	Year with four numbers. For example 2007.

Example
You have the variable <%isodate%> in your object file and you would like to use a format other than the default format. To change the date's format, you could write:

```
<%isodate$d("mmmm d, yyyy")%>
```

If the value for <%isodate%> is 20010601, this would result in the date being written like this:
June 1, 2001

Formatting text

You may want to use a different formatting of a field, rather than the default supported by the variable. To limit the number of variables the object files must support, it is possible to specify the formatting directly in the pbo file. You cannot use these functions on fields that include other object files.

Syntax

The syntax is as follows:

```
<%VarName$<operator>("<format>")%>
```

It should be noted, that it is possible to have several $ operators on a variable. These are executed from left to right.

$	The delimiter that tells the system that formatting for the variable is supplied.
<format>	This field is used to format the variable.
<operator>	Tells the system which class of formatting should be used. The operator is case sensitive. Only the formatting options are listed here. There are other operators available for extracting and replacing text. Available operators are:

- **$a** is used to generate HTML links of URL's and email addresses found in non-HTML text.
  Syntax is

  ```
  <%VarName$a[(Target[,
  URLLinkClass[,
  EmailLinkClass]])]%>
  ```

  and it will produce either

  ```
  <a href="URL" class="URLLinkClass"
  target="Target">URL</a>
  ```

  or

  ```
  <a href="mailto:EMAIL"
  class="EmailLinkClass">URL</a>
  ```

- **$e** is used to replace ASCII characters 161 to 255 with HTML entities.
- **$j** is used to convert text from plain text to URL encoded text.
- **$J** is used to convert text from URL encoded to plain text.
- **$h** is used to convert the variable from text to HTML.
- **$H** is used to convert the variable from HTML to text.
- **$l** is used to convert the variable to lowercase. For example, <%Superheading$l%> will show as 'this is a test'
- **$n** is used to convert the first letter of every word to uppercase (Name case). For example, <%Superheading$n%> will show as 'This Is A Test'. It will also convert all the other characters in each word to lowercase, so BMW will become Bmw.
- **$u** is used to convert the variable to uppercase. For example, <%Superheading$u%> will show as 'THIS IS A TEST'
- **$p** is used to convert the variable to Norwegian proper case. For example, <%Superheading$p%> will show as 'This is a test'
- ***$b*** *is used to convert the variable to US proper case. (not supported)*
- **$t** is used to strip out all HTML codes. Optional exclude list as parameters (comma separated) <%Summary$t("br","b","u")%>
- **$o** converts date to text in FAST InStream search compatible format.
- **$O** converts datetime to a text in FAST InStream search compatible format.
- **$C** Surrounds variable contents in CDATA tags. Use (F) to force the tags if no special characters are present. For example: <%Title$C(F)%>
- **$s** Returns the length of the variable
- **$D** Converts ISODateTime into Extended format. For example: <%ISODateTime$D%> will output in a GMT minus local time offset, like this *Wed, 13 Feb 2008 16:16:00 -0400*

	• **$S** Runs a script, which name is given as the input parameter. The second paramter given is the script language. The second parameter can be left out if the script has a .vbs extension or by adding the file extension to the name. For example: <%currentdomain$S("myScript")%> will call the script myScript.vbs. Both <%currentdomain$S("myScript","JScript")%> and <%currentdomain$S("myScript.js")%> will call the script myScript.js. The script must be places in the scripts folder of the templates directory. The name and value of the variable the script is executed on (in the previous example, this is currentdomain) is avaliable through `SOSE.GetVar("_Variable")` and `SOSE.GetVar("_VariableValue")`. `(From version 6.9.0sr12): To suppress all errors add a third param, with the value 0, example: <%currentdomain$S("Script.php5", "PHPScript", "0")%> default is show errors.` • **$E** performs date calculation on a date using ISO format (YYYYMMDD or YYYYMMDDHHNN or YYYYMMDDHHNNSS). You can run $d after the calculation to format the new date. The value passed is in seconds. Positive and negative values are allowed. For example, <%ActiveDate$E(86400)$d(yyyy/mm/dd HH:NN:SS)%> adds a day to the active date. (There are 86,400 seconds in a day). <%ActiveDate$E(-86400)$d(yyyy/mm/dd HH:NN:SS)%> subtracts a day from the active date.
VarName	The name of the variable.

Formatting numbers

This function is used to format numbers output by the system. You can select the desired number of decimals, separators, and so on. The function is triggered by `$f,` like this:

```
<%Amount$f("#,##0.00")%>
```

The following table shows how the format affects different values.

Format	1234	-1234	0.5	0
0	1234	-1234	1	0
0.00	1234.00	-1234.00	0.50	0.00
#.##	1234	-1234	.5	
#,##0.00	1,234.00	-1,234.00	0.50	0.00
#,##0.00;(#,##0.00)	1,234.00	(1,234.00)	0.50	0.00
#,##0.00;;Zero	1,234.00	-1,234.00	0.50	Zero
0.000E+00	1.234E+03	-1,234.00E+03	5.000E-01	0.000E+00
#.###E-0	1.234E3	-1.234E3	5E-1	0E0

Example

```
<td>Price:</td><td>$<%Amount$f("#,##0.00")%></td>
```

The system will output the correct thousand and decimal separator based on the local setting on your FTP server.

Multiple formatting operations

You can also use multiple formatting operations on the same variable.
<%Date$d="dd. mmmm yyyy"%> will output "16. May 2001".

Output the month in lowercase by combining d and l (lowercase) like this:
<%Date$d("dd. mmmm yyyy")l%> you will get "16. may 2001".

Notice that I must be last, because the commands are executed in the order they are listed. The commands must be placed in front of the formatting information (="dd. mmmm yyyy") and only one command that requires formatting can be used.

Arithmetic

You can add, subtract, multiply, divide and perform the modulo operation with the values in variables. This is only possible for the variables that contain an integer. An integer is a number without decimals.

Syntax

```
<%VarName#[+,-,*,/,%]<value>%>
```

If you do not specify what kind of operation you are performing, the number will be added.

Varname	The name of the variable.
#	Tells the system to perform an operation on the value.
+	Tells the system to add the value to the variable.
-	Tells the system to subtract the value from the variable.
*****	Tells the system to multiply the value with the variable.
/	Tells the system to divide the variable by the value.
****	Used to check if a variable is divisible without leaving a fraction. If this is true, the value will be true. You use it like this: `[%{Counter\4}This text is output if counter is divisible by 4 without a fraction%]`

%	The % tells the system to perform the modulo operation, which is the operation of finding the remainder in division. For example, 10 modulo 4 = 2 because when you divide 10 by 4, 2 is the remainder.You can also use the modulo operation to output different code based on the remainder. Entering `{[Counter%3]=1}` will be true when the counter has the value 1, 4, 7 etc. This allows you to make a three column output like this:
	`[%{[Counter%3]=1}<tr>%]`
	`<td>Content</td>`
	`[%{[Counter%3]=0}</tr>%]`
	The table row start and end will only be output at the start and end of the line.

<value>	This is the number you would like to use in the operation.

Examples
In this example, the field `<%Width%>` contains the number 240:

```
<%Width#8%> = 248
<%Width#+8%> = 248
<%Width#-8%> = 232
<%Width#*2%> = 480
<%Width#/2%> = 120
<%Width#%2%> = 0 (modulo is the remainder of a division)
```

Arithmetic and formatting

You can combine arithmetic and formatting of variables. The calculation must be done first.

<%CashFlow#/2$f("#,##0.00")%> will apply thousands separators after dividing the value of the variable by two. If <%CashFlow%> containe d the value "2432", the result will be "1,216.00".

Variable check

You can check if a variable contains a text value. The match looks at the whole variable. You cannot use this function to match a substring. You can check for a substring using a regular expression.

Case insensitive	`[%{Name=Bob}%]` will output something if the name is Bob. This check is case insensitive.
Case sensitive	By typing `[%{Name==Bob}%]` the check is case sensitive.
Quote marks	If you are going to check a string that contains quotation marks, you must prefix the quotation marks with back slashes like this: `[%{Name="\"Fast\" Willy Parker"}%]`
Special characters	You can check for values containing spaces or the special characters & and | by using quotation marks around the string like this: `[%{Department="Research & Development"}%]`

Extracting a substring

This function allows you to extract a substring of a variable. There are three different substring functions.

Character	Use the option like this:
	`$c(start,count)`
	Start is the character with which you start, count is the number of character you would like to show. If you leave the count blank, it will show the rest of the string. For example:
	`<font size="5"><%Heading$c(1,1)%></font><%Heading$c(2)%>`
	This example will make the first letter in the heading bigger.

Line	Extracts lines from a multiline text. Use the option like this:
	`$L(LineStart,LineStop)`
	LineStart is the line with which you start. LineStop is the line with which you stop.
	`$l`
	converts the text to lowercase.

Word	Copies a substring and then truncates the last character or word in the text to ensure it ends with a whole word. Use the option like this:
	`$w[(StartPos[,count])]`
	Startpos is the character with which you start, count is the number of characters you would like to show. If you leave it blank, it will show the rest of the string. This function does the same as the
	`$c`
	option, but it does not break words.

Length check

You can also check the length of a text string. The basic test is to check the value of a string. By adding [ ] (square brackets) around the field, you can check the length instead.

Example:

You would like to shorten titles so they stay on one line on the front page. In addition, you would like to add ... to show that the title is longer. If the title is shorter than 50 characters, the whole title will be shown without the dots.

```
[%{[Title]>50}<%Title$c(1,47)%>...%|%<%Title%>%]
```

This test will cut titles longer than 50 characters at 47 characters and add three dots.

URL replace

This function allows you to make internet addresses in text fields into hyperlinks. The function will automatically convert all addresses starting with http:// ftp:// and www. into hyperlinks. All e-mail addresses will be converted into `mailto:` links.

To use this function, add $a to the field on which you would like to use the function. You cannot use this function on a field that calls another object file.The body field in ArticleParagraph.pbo looks like this with the function added:
<%Body$a%>

Options

You can set the target and classes for the links. This is done by adding the following options:
$a(target,urlclass,mailclass)

All parameters are optional, so you can write:
$a, $a(_top), $a(_top,extlink,maillink) or $a(,extlink).

Replace text

This function allows you to replace text in a variable. The function is case insensitive. (For case sensitive use $R("saxotech=SAXOTECH"))

For simple replace, that does not include regular expressions, consider using $x which will perform better than $r, if used without the r+ modified.

Example 1	You can use the function like this:<%Body$r("Banana=Apple")$r("Orange=Lemon")%>This function will replace every occurrence of the word banana with apple and every occurrence of the word orange with lemon.
Example 2	You can also use the function to change the way a word is written:<%Body$r("saxotech=SAXOTECH")%>This function will ensure that SAXOTECH is always written in uppercase letters.
Example 3	The quotation mark is a special character and must be doubled up for the system to understand it. If you want to replace a double quotation with a single quotation you can write it like this:<%Body$r(""""="")%>
Example 4	The backslash is a special character and must be doubled up for the system to understand it. If you want to replace a double backslash with a single backslash you can write it like this:<%Body$r("=backslash")%>Backslash is used for specifying regular expressions. Please see the **Regular expresions** section below.
Example 5	Use comma to use multi replace <%Body$r("Maj=May","Juni=Jun")%>
Example 6	[%{Imagelink}<thumbnail><![CDATA[<%Imagelink%>&MaxW=150&<%Heading$x(" ","-")x("æ","ae")x("ø","oe")x("å","aa")jx("\%..","-",r+)l%>.jpg]]></thumbnail>%]

Regular expressions

Regular expressions can be used in the replace function. This expands the basic replace function.

Finding documentation	You can find the documentation of **regexp** on the web by searching for the phrase "perl regexp". There are also books available on this subject.
Matching text	There are several functions for matching text: • **.** (period) Matches any single character except a new line character. • * Matches the preceding character zero or more times. • **+** Matches preceding character one or more times. • **^** Matches the beginning of the line. • **$** Matches the end of line. • **[xyz]** Matches a character set. Matches any characters between brackets. • [^xyz] Matches a negative character set. Matches any characters NOT between brackets. • \| Used for creating OR statements (expression) Brackets an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression. • * Marks the next character as a special character. "w" matches the character "w". "\w" matches any character. The opposite is also true. • **$** Matches the end of the line "$" matches the dollar sign. • **\w** Matches one alphanumeric character. • **\W** Matches any non alphanumeric character. • **\d** Matches one digit. • **\t** Matches a tabulator. • **\r** Matches carriage return. • **\n** Matches new line. • **\s** Matches any white space character. For example: space, tabulator, carriage return and new line.
. (period)	Matches any single character except a new line character.
*****	Matches the preceding character zero or more times.

+	Matches preceding character one or more times.
^	Matches the beginning of the line.
$	Matches the end of line.
[xyz]	Matches a character set. Matches any characters between brackets.
^xyz	Matches a negative character set. Matches any characters NOT between brackets.
| (pipe)	Used for creating OR statements
(expression)	Brackets an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression.
****	Marks the next character as a special character. "w" matches the character "w". "\w" matches any character. The opposite is also true. "$" matches the end of the line "/$" matches the dollar sign.
\w	Matches one alphanumeric character.
\W	Matches any non alphanumeric character.
\d	Matches one digit.
\t	Matches a tabulator.
\r	Matches carriage return.
\n	Matches new line.
\s	Matches any white space character. For example: space, tabulator, carriage return and new line.
. (period)	Matches any single character except a new line character.
*****	Matches the preceding character zero or more times.
+	Matches preceding character one or more times.
^	Matches the beginning of the line.
$	Matches the end of line.
[xyz]	Matches a character set. Matches any characters between brackets.
[^xyz]	Matches a negative character set. Matches any characters NOT between brackets.
*****	Used for creating OR statements
(expression)	Brackets an expression to use in the replace command. A regular expression may have up to 9 tagged expressions, numbered according to their order in the regular expression.
****	Marks the next character as a special character. "w" matches the character "w". "\w" matches any character. The opposite is also true. $ matches the end of the line "\$" matches the dollar sign.
\w	Matches one alphanumeric character.
\W	Matches any non alphanumeric character.
\d	Matches one digit.
\t	Matches a tabulator.
\r	Matches carriage return.
\n	Matches new line.
\s	Matches any white space character. For example: space, tabulator, carriage return and new line.

Special characters	SAXOTECH Online uses = (the equal sign) as a separator. If you want to search for =, you must use the code **\x3D** for =. Quotation marks must doubled like this: " " ", because the first one signifies that the next character is a special character.
Syntax	`$r` triggers the use of regular expressions:<%Body$r("Bob=Robert")%>
Turning off case insensitive	The options "case insensitive" and "greedy" are turned on by default. You can turn these functions off by writing `(?-i)` and `(?-g)` at the beginning of the expression.
Variables	The following variables are available:$0 contains the whole matching string.$1, $2 etc, will contain the parts if the expression is split up with parenthesis. In the function $r("(\w+)Orange(\w+)=$1 Lemon $2", which is a more complex way of replacing text, $0 contains the whole variable, $1 contains the text in front of Orange and $2 contains the text after Orange.

Examples

Matching text

This matches one or more alphanumeric characters:

```
\w+
```

This matches Smith or Jones:

```
Smith\|Jones
```

This matches the entire string:

```
^.+$
```

This matches "man", "men", "min" but not "moon":

```
m.n
```

This matches "test", "teest", "teeeest" etc. It does not match "tst":

```
Te+st
```

This matches "test", "teest", "teeeest" etc. AND "tst":

```
Te*st
```

Adding text

Use this function to add text. In the example the string "Alice, Bob and Charlie went on a road trip" is changed to "cool Alice, cool Bob and Charlie went on a road trip":

```
$r("(Alice | Bob)=cool $0")
```

When the system finds Alice or Bob, it will replace the word with the text on the right hand side.

```
$0
```

is the matching text.

Changing text

The quotation mark is a special character and must be doubled up for the system to understand it. If you want to replace a double quote with a single quote you can write it like this:

```
<%Body$r("""=")%>
```

You can also change text in the string. There are multiple ways to do this:

Example a
The matching word is replaced with another word.

```
<%Body$r("Bob=Robert")%>
```

Example b
This code will change "We see that x=4" to "We see that x equals 4". Here we are looking for the = sign. Since this is a special character we must look for the hex value instead. In addition to replacing the equal sign, a space is inserted before and after the text.

```
<%Body$r("\x3D= equals ")%>
```

Example c

```
<%Body$r("(\w)\x3D(\w)=$1 equals $2")%>
```

The equal sign will only be replaced if an alphanumeric character is next to the equal sign on both sides.

```
$0
```

contains "x=4" in this example.

Removing text

To remove the text (unknown address) from the story byline you can use this function:

```
<%BylineEmailPart$r("(unknown address)=")%>
```

The text matched on the left hand side is not replaced, so the text is removed from the string. The backslashes in front of the parenthesis are there so they are treated as characters and not operators.

Extracting text

You can also extract a part of a string.

```
<%currentrequest$r("(^.+category\x3d)(\w+\d*)(.*)=$2")%>
```

The first parenthesis looks for text from the beginning of the line to category=. The second parenthesis looks for x amount of letters and zero or more numbers. The third parenthesis looks for the rest of the string. The second parenthesis will contain the category ID because it will find all text and numbers between = and &.The function as used here complicates the problem. <%Currentrequest%> is only available in a macro, which splits all URL parameters into their own fields. The field <%category%> should be used instead, since it contains the same information.

Wrapping text

You can also match the entire field to output text before or after the field. Since . does not match new line, every line in the field will be wrapped in paragraph tags.

```
<%Body$r("(?-g)^.+$=<p>$0</p>")%>
```

This function will put every line output inside \ . "Greedy" has been turned off so each line is wrapped instead of the whole body field.

Substring match

This function can be used for checking if a field contains a text string and outputting the text "Order print" if it does.

```
<%photo$r("(^.+)staff(.+)=1")V(sh_staffphoto)%>
[%{sh_staffphoto=1}Order print%]
```

The function sets a flag if the text "staff" is in the photo credit field. There is then a test to see if the flag is set to output the text. A temporary variable is used for this. Please see Writing new variables below for more information on temporary variables.

Extended replace

Extended replace is another replace function. The difference between regular expressions and extended replace are subtle. The big difference is that the extended replace function can use variables inside the expression.

Syntax

$x triggers the use of regular expressions:

```
<%Body$x("OldText","NewText"[,"ReplaceFlags"])%>
```

The Replace flags parameter is optional.

Replace flags

Replace flags are used to control how the replace function should behave. The flags are switched **on** using + and **off** using using - in the text string, written as the third parameter in the command. Every flag must have either a + or - specified immediately after the flag itself. For example:

```
<%Body$x("OldText","NewText","r+i+g-m+")%>
```

These options will turn regular expression mode on, ignore case on, greedy mode off and multi line mode on.

Regular expression flags

These extra flags are available when you are using regular expressions.

- **g**: Greedy mode. On by default.
- **m**: Multi line mode. Off by default. When this function is turned on, the ^ symbol will match every line beginning and $ will match every line end. When this function is turned off, the ^ symbol will match the beginning of the text and $ will match the end of text.
- **s**: Single line mode. Off by default. When this function is turned on, . (period) will match any character including ^ and $. When this function is turned off, . (period) will match any character except ^ and $.

Standard flags

- **i**: Ignore case. Off by default.
- **a**: Replace all occurrences. On by default. This flag has no effect when using regular expressions. When using regular expressions, all occurrences will always be replaced.
- **r**: Use regular expression. Off by default. If "r+" is included in ReplaceFlags, the "NewText" is treated as a regular expression substitution template. This means $0 will return the expression match and $1..$n will return the sub expression matches.

Special characters

In both OldText and NewText, quotation marks (") must be escaped like this: \". And consequently backslashes must be escaped like this:

```
```

If these special characters are not escaped, the function will not work.

Variables

Any variable value can be used in OldText and NewText. This is done by joining static text and a #Variable# expression with '+', or using #Variable# expressions alone.If you have the summary in the variable <%summary%> with the value "The writer believes this is wrong." and you have the byline in the variable <%Byline%> with the value "Bob Toner", then the expression

```
<%summary$x("The writer", #Byline#)%>
```

will output "Bob Toner believes this is wrong".

Examples

Adding text

This will find all occurrences of "saxotech" in <%Body%>, regardless of case setting, and replace them with "SAXOTECH" in yellow text.

```
<%Body$x("saxotech", "<font color=\"Red\">SAXOTECH</font>", "i+")%>
```

Adding to a variable*

This function adds on the value of counter when the counter value is found in the text.

```
<%number$x(#Counter#+": ", #[Counter+1]#+": ")%>
```

Adding variable

If the <%\Title%> field has the value "The team wins again and again" and the <%Team%> variable has the value "BMW Williams", the function will output "The BMW Williams team wins again and again".

```
<%Title$x("The team", "The "+#Team#+" team")%>
```

Add numbers

Start = 3, Count = 2 will output 5.

```
<%Start#+#Count#%>
```

Multiple replaces*

If you want to use multiple replace function on the same variable, you can write multiple replace commands.

```
<%field$x("apples", "oranges")x("cars", "trucks")%>
```

Regular expression

This will put every line output of <%Body%> inside and .

```
<%Body$x("^.+$", "<p>$0</p>", "r+m+g-")%>
```

Replace with variable

The function replaces the text yours truly with the value from the Name variable.

```
<%Body$x("yours truly", #Name#)%>

Another more complex example could be.

<!-- Define a local variable, in this case a string in the URL -->
<%?myVar=mydomain%>
<%CurrentUrl$x("(^.+)"+#myVar#+"(.+)","1","r+")V(custom_myvar)%>
[%{custom_myvar=1}
  URL before replace: <b><%CurrentUrl%></b><br />
  URL after replace: <b><%CurrentUrl$x(#myVar#, "replacestring")%></b><br />
  <a href="<%CurrentUrl$x(#myVar#, "replacestring")%>"><%CurrentUrl$x(#myVar#,
"replacestring")%></a>
%|%
  No such string in the URL
%]
```

Replacing text

This will find all occurrences of "apples" in <%Body%> and replace them with "oranges".

```
<%Body$x("apples", "oranges")%>
```

Writing new variables

You can write the result of a function to a new variable using the V option in the substring match example:

```
<%photo$r("(^.+)staff(.+)=1")V(sh_staffphoto)%>
```

Use a new variable called

```
sh_staffphoto
```

It is possible to overwrite system variables. Therefore, all your variables should be prefixed with **sitecode**_. If you overwrite system variables, there might be severe consequences.

Local variables

To create a local variable, prefix with ?:

```
<%?sh_myvariable%>
```

This variable is only available in the current macro. For example:

To set a local variable, "sh_foo" with the value "10":

```
<%?sh_foo=10%>
```

To activate this variable:

```
<%sh_foo%>
```

It is possible to overwrite system variables. Therefore, all your variables should be prefixed with **sitecode**_. If you overwrite system variables, there might be severe consequences.

Global variables

To create a global variable, prefix with ?@:

```
<%?@sh_myvariable%>
```

The value is also available for other macros used on the page. For more information, please see Global variables. For example:

To set a global variable, "@sh_bar" with the value "12":

```
<%?@sh_bar=12%>
```

To activate this variable:

```
<%@sh_bar%>
```

It is possible to overwrite system variables. Therefore, all your variables should be prefixed with **sitecode**_. If you overwrite system variables, there might be severe consequences.

Odd and Even variable picture handling

You can use the counter variable to alter the layout of photos in the list of stories.

In a table

Example code

```
[%{Imagelink}

<table align="%][%{Counter is even&ImageLink}right%][%{Counter is
odd&ImageLink}left%][%{Imagelink}" bgcolor="#CCCCCC" cellspacing="0" cellpadding="2"
border=0>
<tr><td><img src="<%Imagelink%>&MaxW=100" width=100 border=0 hspace=0
vspace=0></td></tr>
</table>%]
<font face=Verdana size=2><b><%Title%></b></font><br>
<font face=Verdana size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana size=1>
<a href="<%Link%>">Whole story</a> |
<a href="<%pbcs.dll%>/section?Category=<%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

Example news list

Topstory Variable

The Topstory variable is set to 1 when the priority for the story is from 1 to 3. You can use this variable to set the size of the title.

NewsItem.pbo	With this NewsItem, the story will have the title size 4 for stories with priority 1 through 3 and title size 2 for stories with priority 4 or lower.

```
[%{Imagelink}
<img src="<%Imagelink%>&MaxW=100"
width=100 border=0 hspace=0 vspace=0
align="%][%{Counter is
even&Imagelink}right">%][%{Counter is
odd&Imagelink}left">%]
<font face=Verdana [%{TopStory}
size=4> %|%  size=2> %]
<b><%\Title%></b></font><br>
<font face=Verdana
size=1><%Leadin%></font><br>
<div align="left"><font face=Verdana
size=1>
<a href="<%Link%>">Whole story</a> |
<a
href="<%pbcs.dll%>/section?Category=<
%Category%>">
<%Categoryname%></a></font></div>
<br>
</font>
```

Example news list

pbs macro

<pbs:macro>

Macros are functions related to object script. They allow you to build tags or output other code, based on URL parameters and options in the tag.

Template
This function can be used on all templates.

Folders
The system will look for the macro file in the following folders:
/<templates>/macros/<categoryid>
/<templates>/macros/<maincategoryid>
/<templates>/macros

Caching
You can run the tag uncached, so that it is cached independently of the template itself. When you write `<pbs:!macro.....>`, a separate cache file will be created. You must then use the cache time option to set the refresh interval. Note that this is the behavior when the macro file contains another <pbs> tag inside of it to be rendered. If no <pbs> tag gets executed as a result of there not being one present or objectscript conditionals, then the macro tag will NOT be cached and instead be evaluated on every request. This is true whether or not the cachetime parameter is set in the macro tag.

Options

Before and After	The Before option makes sure that the tag is parsed before the story, or in whatever context you are using the macro. It is used like this: `<pbs:macro before name="macro">` Conversely, the After option makes sure that the tag is parsed after.
Cachequalifiers	This option is used for specifying which fields are needed to make the cache file unique so another page does not overwrite it with different content. The option is used like this: `<pbs:!macro cachetime=1440 CacheQualifiers="Date,Category,foo,site,ArtNo" name="macro">` This function can take fields from the URL and use them to create unique file names. From version 6.6.0sr43, 6.8.0sr8 if the system variable @StoryId is used as cachequalifier then the system will consider that macro to be part of the story cache files and the cache file will be invalidated with the rest of the story cache files when nocaching a story. **Custom CacheQualifiers** In order to use custom cachequalifiers they must be added to the SectionActionCacheQualifier of the [CACHE] section on the server side publicus.ini. The list should be comma delimited. For example: SectionActionCacheQualifier=foo,bar
Cachetime	This sets the time the include should be cached before it is refreshed. This option is only used when the tag is running uncached. `<pbs!macro name="header" cachetime="60">`
IgnoreNocache	The ignoreNocache=1 parameter on the macro tag can be used to indicate that the content of the cache file is static within the cache time. It will not be rebuild when doing nocache=1.

InlineObject	This option allows you to use the functionality of a macro without having to create a file for the macro. Instead, the code that would go in the macro file is set as the value of this option. All variables that would be available in the macro are available here. As a best practice, this option should only be used for very simple macros. Stuffing too much code into the InlineObject option makes functionality difficult to maintain. When InlineObject is used, the Name option is not necessary. Example: `<pbs:macro inlineobject="<%currentrequest%>">`
KeepEmptyLines From version 7.0.0sr11	The KeepEmptyLines="1" parameter tells the system to not automatically delete lines that are blank. Default is that the system deletes empty lines.
Name	This option defines the file name for the macro. If you set the name to AP, the system will look for the macro file AP.pbo.
Dump	Will save a logfile with all parameters in the DBLog folder. `<pbs:macro name="header" dump="1">`
Noparse	This parameter should be used lightly. In the most extreme of circumstances, within a specific section of a website, it might be required to call a macro that then calls two or more other macros, which require the sharing of created global variables. Within this subsection of a website, where the two or more macros will need to share user-created variables, the noparse option should be added to the two or more second-level macro tags. This parameter ensures that the bottom level macros execute simultaneously and that any user-created variables on the same level are shared
Other options	You can define other variables by adding `field=value` to the tag. In addition, all options specified in the URL will be available in the macro file.When defining other variables, the names of the variables should not be words used elsewhere in SAXOTECH Online for field names, tag names, and so on.
Other variables	For fields are available in addition to system variables, URL, and global variables. If you put a macro on the result page, all fields in the form are available. StaticValues from localdata.ini are available as well.

URL parameters	All parameters in the URL are available as fields. If you have the following URL:
	`/apps/pbcs.dll/article?Date=20020907&Category=NEWS02&ArtNo=221004&Ref=AR`
	You have these fields:
	`<%Date%>`
	with the value 20020907
	`<%Category%>`
	with the value NEWS02
	`<%Artno%>`
	with the value 221004
	`<%Ref%>`
	with the value AR

System variables
The following system variables are available, in addition to variables defined in the tag and taken from the URL. Please see System variables for more system variables.

<%CurrCategoryName%> From version 6.9.0sr5	Holds the current category's display name.
<%CurrMainCategory%>	This field holds the current main category. It is written:`<%CurrMainCategory%>`
<%CurrMainCategoryName%> From version 6.9.0sr5	Holds the current main category's display name.

You can use your own variables in a format like the following:

```
<pbs:macro name="saxo" variable1="text1" variable2="text2">
```

Example 1
You want macro variables to contain the same values as the tags **<pbs:categoryname main>** and **<pbs:categoryname>.**

Use a macro tag like this:

```
<pbs:macro name=blah categoryname="<pbs:categoryname>">.
```

Then define a variable **<%categoryname%>** in the macro that is formatted with object script.

Example 2
You have the following tag on your front page:
<pbs:macro name=AP Lineup=HOME>

This will make the system look for the file AP.pbo, and the field Lineup in the object file. The lineup field has the value HOME.

The macro

```
<script language="JavaScript">\\
 site="OHTOL";section="[%<%Section%>%\|%<%Lineup%>%|]";\\
 </script>\\
 <script language="JavaScript"\\
 src="http://customwire.ap.org/lineups/<%Lineup%>.js">\\
 </script>
```

The first script function sets the variable's site and section. These variables are going to be used by the included JavaScript. Section will be set to either the section or the lineup field, depending on what is used in the macro tag.The second line loads a JavaScript based on the lineup variable settings.

Example 3
The `<pbs:macro>` tag can be used for many things. For example, on election night when AP sends results in XML format, you can have a page with election results:

```
<pbs:\!macro name="Election2008" Cachetime="15">
```

- Election2008.pbo

Election2008.pbo

ElectionScript contains some VBScript, which retrieves the election results from AP's FTP server.

In this way, election results from AP's elections service are updated every fifteen minutes.

```
<pbs:script name="ElectionScript" objectclass=1 Language="VBscript" file="xyz.xml">
```

Position-based design

Position-based design allows you to design the layout of each item in a newslist. To create object files for position based design, you add the position number to the end of the object file name. The object files for object class 1 must be named:

NewsItem1.pbo	Used for stories which have no position-based design.
NewsItem1_1.pbo	Used for position 1.
NewsItem1_2.pbo	Used for position 2.
NewsItem1_3.pbo	Used for position 3.

Session function

Use this to read from session files:

```
[BROWSER]
NS=0
IE=1
Version=5
UserAgent=Mozilla/5.0 (Windows; U; Windows NT 5.1; da; rv:1.9.2.6) Gecko/20100625
Firefox/3.6.6
```

```
<pbs:!#sessionvariable# section="BROWSER" name="IE">
```

Will return value "1"

System variables

System variables are available in all object files. They are used to extract a query parameter from the URL for use in an object file or macro. This is done using a regular expression:

```
<%currentrequest$r(".*myParam\x3d(\w+).*=$1")V("myVariable")%>
```

The example above takes the value of the query parameter myParam and stores it in an objectscript variable called myVariable.

<%Activedate%>	Contains the publishing date.
<%ClientIP%>	Contains the IP address of the computer that generated the page.
<%CharSet%>	Contains the ICharset like iso-8859-1.
<%CurrCategory%>	Contains the current category ID. It must be run within a macro when used in article object files.
<%CurrCategoryName%>	Holds the current category's display name.
<%CurrMainCategory%>	Contains the current main category ID.
<%CurrMainCategoryName%>	Holds the current main category's display name.
<%CurrProfile%>	Contains the current profile ID. It must be run within a macro when used in article object files.
<%CurrentDate%>	Contains the current date.
<%CurrentDomain%>	Contains the domain from the Publicus.ini.
<%CurrentRequest%>	The request part of the URL. It is written: `<%CurrentRequest%>` Currentrequest returns the rewritten (after modified by URLReWrite) version of the URL, including nocache=1 if it is present.
<%CurrentRequestWithoutNocache%>	Currentrequest returns the rewritten (after modified by URLReWrite) version of the URL, without nocache=1.
<%CurrentRequestUnmodified%>	This variable returns the value before it is processed by URLRewrite while <%currentrequest%> returns the rewritten (after modified by URLReWrite) version of the URL, without nocache=1.
<%CurrentRequestMethod%>	Request method for the page. It is written: `<%currentrequestmethod%>`
<%CurrentUrl%>	Contains the current URL.

<%inheritedlink%>	Contains the same value as
	`*<%currentrequest%>*`
	It is written:
	`<%inheritedlink%>`
	Inheritedlink will exclude the query parameter nocache=1.

<%ImageServer%>	Contains the value from the image server setting in Publicus.ini.
<%@StorySite%>	Contains the site (XX) from a story action
<%@StoryDate%>	Contains the publish date (yyyymmdd) from a story action
<%@StoryCategory%>	Contains the category from a story action
<%@StoryId%>	Contains the Story ID (Lopenr) from a story action
<%StaticImagePath%> **From Version 7.1.0sr6**	Contains the link to use in front of Image URLs to link them as static images through PbcsI.dll. (/apps/pbcsI.dll/staticimage)
<%@IsCrawler%> **From Version 7.1.0sr13**	Returns [0|1] based on apps_pbcsdb.ini [CRAWLERS] section
<%@UserAgent%> **From Version 7.1.0sr13**	Return the Browser useragent
<%@UserAgentGroup%> **From Version 7.1.0sr13**	Return the name of the user agent group defined in rewrites.ini [AGENTS]
<%@Referer%> **From Version 7.1.0sr13**	Return the HTTP Referer value

Profile variables:

<%Profile_id%>	Contains the ID of the current profile
<%Profile_Name%>	Contains the name of the current profile
<%Profile_Guid%>	Contains the guid of the current profile
<%Profile_parentid%>	Contains the if of the parent to the current profile
<%Profile_priority%>	Contains the priority of the current profile
<%Profile_Fields_[fieldname]%>	Contains the value of the extra field given by [fieldname] of the current profile.
<%Profile_Category%> From versions 7.1.0sr13 and 7.2.0sr2	Contains the category ID associated with the current profile.
<%Profile_URL%> From versions 7.1.0sr13 and 7.2.0sr2	Contains the URL associated with the current profile.

Menu Tags

The menu tags are ways of building navigation using templates and objects rather than building each link manually. You can also put in tags manually where you want menu links. When you use the menu tags, you can turn entries on and off using settings in the category setup.

- pbs categorytree
    - CategoryTree.pbo
    - CategoryTreeItemActive.pbo
    - CategoryTreeItemPassive.pbo
    - CategoryTreeItemSeparatorColl.pbo
    - CategoryTreeItemSeparatorExp.pbo
    - CategoryTreeItemSeparatorNone.pbo
    - CategoryTreeItemSubTreeIndention.pbo
- pbs ft
- pbs leftmenu
- pbs menu-image
- pbs menu-text
- pbs profiletree
    - ProfileTree.pbo
    - ProfileTreeItem.pbo
    - ProfileTreeItemActive.pbo
    - ProfileTreeItemPassive.pbo
    - ProfileTreeItemSeparatorColl.pbo
    - ProfileTreeItemSeparatorExp.pbo
- pbs qwiknavsearch
    - QwikNavSearchForm.pbo

pbs categorytree

<pbs:categorytree>

This tag allows you to build a category tree showing all categories. This tag can be used on all templates.

Options

ExcludedCategoryList	Specifies the categories that the tag should not list. By default, this option is not used. To exclude a list of categories, enter:
	`ExcludedCategorylist=SPORTS01,SPORTS02,NEWS01,NEWS02,NEWS03`
	Wildcards can also be used:
	`ExcludedCategorylist=SPORTS%,NEWS%`

Expand	Controls which branches should be expanded. The expand option has the following values: • **All** - All branches are expanded. • **Active** - Only the active branch is expanded.
Levels	Specifies how many levels will be expanded by default. Usage: `Levels=3`
Object class	This option sets the object class for the object files.
Shownext	Expands all subcategories when a category is clicked.

Type	This option controls the type of category tree that is built. This option is set by using `type=xxx` The following values are available for xxx: • **upper** - The tree goes from the current category to the top of the tree. • **top** - The tree has all top level categories. • **complete** - A complete tree of all categories. • **lower** - The tree shows the category currently being viewed and its subcategories
Usesectioncat	Makes the function use the category in which the story is shown and not the category in which the story is placed. By default, the function uses the category in which the story is placed. Enter `usesectioncat=1` to use the category to which the profile is attached.

- CategoryTree.pbo
- CategoryTreeItemActive.pbo
- CategoryTreeItemPassive.pbo
- CategoryTreeItemSeparatorColl.pbo
- CategoryTreeItemSeparatorExp.pbo
- CategoryTreeItemSeparatorNone.pbo
- CategoryTreeItemSubTreeIndention.pbo

CategoryTree.pbo

<%CategoryTreeContent%>	Contains the entire category tree.

CategoryTreeItemActive.pbo

Used for the category being viewed.

<%CategoryID%> **From version 7.0.0sr11**	Contains the ID of the category.
<%Name%>	Contains the name of the category.
<%OverView%>	Contains the link to the category. From version 7.0.0sr11 if the site uses URLRewrite, then the output is the format of /section/<categoryid>
<%OverViewWithDomain%> **From version 7.0.0sr15**	Contains the link to the category with live domain.

CategoryTreeItemPassive.pbo

Used for the categories not currently being viewed.

<%CategoryID%> **From version 7.0.0sr11**	Contains the ID of the category.
<%Name%>	Contains the name of the category.
<%OverView%>	Contains the link to the category. From version 7.0.0sr11 if the site uses URLRewrite, then the output is the format of /section/<categoryid>

<%OverViewWithDomain%> **From version 7.0.0sr15**	Contains the link to the category with live domain.

CategoryTreeItemSeparatorColl.pbo

This object is shown in front of all objects that contain a collapsed branch.

<%Expand%>	Contains the action necessary to expand the branch.

CategoryTreeItemSeparatorExp.pbo

This object is shown in front of all objects that contain an expanded branch.

<%Collapse%>	Contains the action needed to collapse the branch.

CategoryTreeItemSeparatorNone.pbo

This object is used in front of objects that cannot expand or collapse. This file has no fields.

CategoryTreeItemSubTreeIndention.pbo

When outputting a category subtree (type is either complete or lower) the indentation will default to three non-breaking spaces. This object file can be used to provide a replacement indentation that will override the default.

pbs ft

<pbs:ft>

This tag was originally made for building the footer menu. The tag has morphed into the de facto tag for building menus. Menus can also be coded by hand using the anchor tag.

Options

The tag supports several options for modifying text style.

Activedot	Used to show which category is currently being viewed. When you use the activedot parameter, the menu must be enclosed in a table because the tag will output a table row for each item in the menu. Example:
	`<table border=0>`
	`<pbs:ft section=FRONTPAGE`
	`activedot="/images/arrow.gif"`
	`dotwidth=7 dotheight=11 nobold>`
	`<pbs:ft section=NEWS`
	`activedot="/images/arrow.gif"`
	`dotwidth=7 dotheight=11 nobold>`
	`</table>`
	The activedot parameter is the path to the image you would like to use.

DotHeight	Sets the height of the activedot.
DotWidth	Sets the width of the activedot.
Font Size	Controls the font size for the menu item. The tag will use the default font size. However, you can use font size from 1 to 6.

Front page	Creates a link to the front page. This option is not used by default. Example: `<pbs:ft frontpage>` When the option is used, the menu link goes to the front page. You can also link from the menu to the front page using the section option. To do this, you must first define the category Front page in the category register and set the link type to front page.

Lbgcolor	Sets the background color of the left cell. If the option is not set, the default background color will be used. This option can be set in two ways:
	
	• To use an image for the background, use the option
	`Lbcolor="img:/images/blue.gif"`
	• To use a color for the background, use
	`Lbcolor="#eeeeee"`

Nobold	Makes the link not bold. The links in the menu are bold by default.
Profile	Defines which profile should be used for the page. The option is set like this: `profile=1012`
Rbgcolor	Sets the background color of the right cell. This option has the same options as the Lbgcolor option. Notice that the option is called rbg and not rgb.
Section	Creates a link to a sectionfront. The parameter is the ID for the category. If the ID for the category is SPORT11 and the name is College basketball, you use `section=SPORT11` as the parameter.

Styleclass	This option is used for setting the style sheet class. The default class is Menu. Setting
	
	`styleclass=0`
	
	will turn off the use of styles. Setting
	
	`styleclass="Main"`
	
	will set the style sheet class to Main.

Target	Sets the target in a frame set.

pbs leftmenu

<pbs:leftmenu>

The Left menu tag is used for building graphical menus. This option is rarely used. The tag will use an image called **lm_categoryname.gif** in the directory **/grafikk**.

The tag supports the following options:

Forside	This option links to the front page.
Oversikt	This option links to a section front. Entering `oversikt=NEWS` links to the news section.

pbs menu-image

<pbs:menu-image>

This tag has the same functionality and options as the **<pbs:leftmenu>** tag. Please see the section for the pbs leftmenu tag for more information.

pbs menu-text

<pbs:menu-text>

This tag has the same functionality and options as the **<pbs:ft>** tag. Please see the section for the pbs ft tag for more information.

pbs profiletree

<pbs:profiletree>

This tag is used for building a tree of all profiles in the system. This tag can be used on all templates.

The tag only supports output of 2 levels in the tree.

Options

Autoexpand	This will make the subcategories expand when a category is selected.
Cachetime	The option cachetime is used to determine how long the tag should be cached. The option is used like this: `Cachetime=X` where X is the number of minutes.
Category	Specifies the category like this: `category="news"` Please note that when you use <pbs:profiletree> as a navigation menu, the category must be specified. If you do not specify the category, the profile tree will be broken when a web guest goes to a category other than the one in the profile tree.
Keepexpanded	Keep the tree expanded

Excludeid	This option is used for excluding profile IDs that should not be part of the tree.Example: `excludeid="1000,1001,..."`
Object class	This option sets the object class for the object files.
Sectioncat	Sets the category in the link.
Topnode	This option is used for setting the top node for the profile tree. This enables you to build a profile tree for just one branch.

This function requires object files.

- ProfileTree.pbo
- ProfileTreeItem.pbo
- ProfileTreeItemActive.pbo
- ProfileTreeItemPassive.pbo
- ProfileTreeItemSeparatorColl.pbo
- ProfileTreeItemSeparatorExp.pbo

ProfileTree.pbo

If the ProfileTree.pbo does not exist, it will default to <%items%>, which will output the same result as earlier versions.

<%TopNode%>	The profile name of the ID. If no TopNode is given, it will be set to Root <%Items%>, which is the list of profile items.
<%TopNodeID%>	The TopNode ID given as input to the tag.

ProfileTreeItem.pbo

<%IsSubNode%>	This field has a value if the item is a sub node.
<%Item%>	Displays the item.
<%Separator%>	Displays the separator.

ProfileTreeItemActive.pbo

<%IsExpandable%>	This field has the value 1 if the item has sub items.
<%IsExtURL%>	This field has the value 1 if the item points to an extra URL.
<%ProfileName%>	Contains the name of the profile.
<%URL%>	Contains the URL for viewing the profile

ProfileTreeItemPassive.pbo

<%IsExpandable%>	This field has the value 1 if the item has sub items.
<%IsExtURL%>	This field has the value 1 if the item points to an extra URL.
<%ProfileName%>	Contains the name of the profile.
<%URL%>	Contains the URL for viewing the profile

ProfileTreeItemSeparatorColl.pbo

<%ExpandURL%>	This URL is used for expanding the branch.

ProfileTreeItemSeparatorExp.pbo

<%CollapseURL%>	This URL is used for collapsing the branch.

pbs qwiknavsearch

<pbs:qwiknavsearch>

This function can be used for jumping directly to a story or a sectionfront. This function is comparable to AOL keywords. This tag can be used on all templates.

To use this function, you must assign QwikNav words to the stories to which you would like to jump. You can also assign QwikNav words to aliases in the Alias database.

This tag has no options.

- QwikNavSearchForm.pbo

QwikNavSearchForm.pbo

<%Action%>	Contains the action for the form.
ErrorUrl	Used to set the error URL, which is used when the QwikNav search word does not exist.
QwikNav	Used to capture the QwikNav word.

The following is an example of the QwikNavSearchForm.pbo:

```
<form action="<%Action%>">
<b>QwikNav:</b><br>
<input type="hidden" name="Errorurl"
value="/apps/pbcs.dll/misc?Url=/templates/QwikNaverror.pbs">
<input type="text" name="QwikNav" size="9"
style="{font-family: Verdana, Geneva, Arial, Helvetica,
sans-serif; font-size: 11px; width: 80px;}"> 
<input type="submit" value="&raquo;"
style="{background-color: #efefef;
border: 1px solid Black;
font: 10px Verdana;}">
</form>
```

Section Front Tags

The section front tags can only be used on the section front templates. These tags are used for building lists of stories. In addition to the section front tags, you can also use the overview tags to build lists of stories. The default section front template is called **gen_ovr.pbs**. For more information on template naming, please see Template names and URLs.

This tag is used for listing stories on section fronts. You should use the pbs newslist tag to build section fronts.

For the category setup set the overview type to 2, if the system should always show the section front, even if there is only one story.

- pbs oversiktnotiser
- pbs sectionfront

pbs oversiktnotiser

<pbs:oversiktnotiser>

This tag is used for listing notices on the sectionfront of a category. Notices are used when you would like to avoid listing very short stories in the main overview. These stories must have the **Notice** checkbox marked in the story properties.

Options

Link	Controls the text in the story link. By default the tag will output [Read more]. Enter `link=????` where ???? is any text of your choice. The system will use this text instead for the link. Setting link=title makes the title the link.
Sortering	There are two possible sort options. By default, stories are sorted to show the newest stories first.Enter `sortoption=prio` to sort the stories according to priority.
Visref	Controls the number of summary characters that will be shown. By default the whole summary will be shown.Enter `leadin=#` to determine how many characters should be shown. This can be any number between 0 and 9999.

pbs sectionfront

<pbs:sectionfront>

This tag is used for listing stories on sectionfronts. You should use the <pbs:newslist> tag to build sectionfronts.

Output options

Count	Controls how many stories should be shown. By default, all stories published on today's date are shown. Enter `count=#` to determine a number of stories to be shown. This option overrides the archive setting in the category register.
Leadin	Controls the number of summary characters that will be shown. By default the whole summary will be shown. Enter `leadin=#` to determine how many characters should be shown. This can be any number between 0 and 9999.
Subcats	Controls whether or not stories from subcategories will be shown in the listing. By default, listing of stories from subcategories is turned off. Enter `subcats=1` to enable listing of stories from subcategories.
Summaries	Determines how many stories should have summaries. By default, all stories have summaries. Enter `summaries=x` to determine that only the first **x** stories will have summaries. Entering any number for **x** will show the summaries for the specified number of stories. Only the title will be shown for the other stories.

Image output options

Image	Controls if a thumbnail picture is shown in the story overview. This is a thumbnail of the main story picture. By default thumbnails are turned off. Entering `image=1` enables thumbnails.
Picturealign	Controls the placement of the thumbnail. By default, thumbnails are placed on the left. Enter `picturealign=right` to change the alignment of thumbnails to the right.

Picturepadding	Sets the padding on the thumbnail. By default, there is 4 pixels of padding around a thumbnail. Enter `picturepadding=#` to adjust the padding around the thumbnail. # can be any number between 0 and 100.
Picturewidth	Controls the width of the thumbnail. By default, all thumbnails are 80 pixels wide. Enter `picturewidth=#` to adjust the width of thumbnails. # can be any number between 10 and 1000.

Formatting options

Cathead	This option controls if the category/subcategory name should be shown above the title. By default this option is turned off. Enter `cathead=1` to enable this option.
Linebeak	Draws a line between each story in the list. By default, no line is drawn. Enter `linebreak=1` to draw a line between each story.
Link	Controls the text in the story link. By default the tag will output [Read more]. Enter `link=????` where ???? is any text of your choice. The system will use this text instead for the link. Setting link=title makes the title the link.

Linkcolor	Sets the link color. By default, the link color is the color set in the <body> tag. Enter
	
	`linkcolor=#????`
	
	to output a font color around the title in the following manner:
	
	`<font color=#????>`

Sortoption	There are two possible sort options. By default, stories are sorted to show the newest stories first. Enter `sortoption=prio` to sort the stories according to priority.

Subcatcolor	Sets the color of the subcategory header. By default, the text color is used. Enter
	`subcatcolor=#????`
	to output a font color around the subcategory name in the following manner:
	`<font color=#????>`

Titlefont	This option sets the font that will be used for the title. The default font face will be used.If you do not wish to use the default font face enter
	`titlefont=????`
	to output a different font face around the title in the following manner:
	`<font face=????>`

Titlesize	This option sets the font size that will be used for the title. By default the size is 4. This will output in the following manner:
	`<font size=4>`
	If you wish to change the font size, enter
	`titlesize=#`
	where # is a value between 1 and 6. This will output a different font size around the title in the following manner:
	`<font size=#>`

Togglecolumns	This option builds a page in which every other line contains two stories.

Article Tags

Article tags are used to display stories.

- pbs article
    - Article.pbo
        - ArticlePhotoGalleryThumbnailItems.pbo
    - ArticleAccessControls.pbo
    - ArticleAccessControlsItem.pbo
    - ArticleByline.pbo
    - ArticleConnectionItem.pbo
    - ArticleConnections.pbo
    - ArticleExtLogin.pbo
    - ArticleExtraPictureItem.pbo
    - ArticleExtraPictures.pbo
    - ArticleFacts.pbo
    - ArticleForumConnectionItem.pbo
    - ArticleForumConnections.pbo
    - ArticleGalleryConnectionItem.pbo
    - ArticleGalleryConnections.pbo
    - ArticleGeoDataItem.pbo
    - ArticleGeoDataItems.pbo
    - ArticleHead.pbo
    - ArticleHotLinks.pbo
    - ArticleLinkItem.pbo
    - ArticleLinks.pbo
    - ArticleLocation.pbo
    - ArticleModified.pbo
    - ArticleMultiMedia.pbo
    - ArticleMultiMediaExtension.pbo
    - ArticleMultimediaLinkItem.pbo
    - ArticleMultiMediaList.pbo
    - ArticleMultiMediaListItem.pbo
    - ArticleMultimediaListItemExtension.pbo
    - ArticleMultimediaListItemXXX.pbo
    - ArticleMultiMediaPlaylistItem.pbo
    - ArticleParagraph.pbo
    - ArticlePhotoGalleryThumbnailItem.pbo
    - ArticlePhotoGalleryThumbnails.pbo
    - ArticlePicture.pbo
    - ArticlePublished.pbo
    - ArticleSummary.pbo
    - ArticleTaxonomy.pbo
    - ArticleTaxonomyItem.pbo
    - ArticleTaxonomyItemPathItem.pbo
    - InvalidArticle.pbo
    - Tips.pbo and articles
- pbs articledate
- pbs articlefield
- pbs article-links
- pbs articlemultimedia
- pbs artikkel-urls
- pbs keywordlinks
    - KeywordLinkItem.pbo
    - KeywordLinks.pbo
- pbs photogallery-links
- pbs themelinks
    - ThemeLinkItem.pbo
    - ThemeLinks.pbo
- Other links

pbs article

<pbs:article>

This tag outputs the whole story or parts of it, depending on which options are used. Using the tag like this will output the story with default formatting. Some of the options on this tag block output of some elements of the story. The tag supports object files.

The article tag is used to display stories. Links to stories are automatically built by the overview tags. It is also possible to access a story by linking directly to it. The link will look something like this:

```
/pbcs.dll/artikkel?Avis=GE&Dato=20001008&Kategori=FRONTPAGE&Lopenr=11008068&Ref=AR
```

The article tag supports the use of article classes. This is a method of predefining different layouts for the article tag using object files. Please see Article classes.

The article tag also supports object files. Please see Object Files to learn more about object files.

The article tag supports the use of object files for controlling layout. You can also use different classes for the object files. Please see Article classes.

This tag is used on article templates. The general template is called **gen_art.pbs**.

Printer-friendly version

The print version of the story is a version that does not have any graphics. The print template is usually used in two different versions: One with pictures and the other without. The code used in the templates is stripped down and uses very little formatting.

The JavaScript in these templates will automatically open the print dialog box if the function is supported. If the function is not supported the text, *To print* *this article open the file menu and choose Print* is shown instead.

The plain text version uses the template **gen_printart.pbs**. The version with pictures uses the template `gen_printpicart.pbs.`. If you are using object files for the layout of stories, you must use the option UseObjects in the article tag on the print templates.

The <pbs:article> has the following options:

Object file options

Objectclass	The object class controls which object files the page will use. By default, the system will use the files in the default object class.Object class can be set from 1 to 999.
Useobject	This option turns off the use of **Article.pbo**. Object files are used for parts of the story instead. This is triggered by entering `useobject=0` in the article tag.

Useobjects	Object files are automatically used if they exist. To turn off the use of object files, add the option
	`useobjects=0`
	in the
	`<pbs:article>`
	tag.

Story selection options

StoryGUID **From versions 6.9.0sr27, 7.0.0sr2**	A GUID of the story to display. If omitted the story will be identified based on the URL parameters.

Output options

Byline	Controls the output of the byline. The default is
	`Byline=1.`
	Enter
	`Byline=1`
	to enable output of the byline. Enter
	`Byline=0`
	to disable output of the byline.

Facts	If the story contains a fact box, setting
	`Facts=1`
	will output the fact box. The default is
	`Facts=0.`
	Enter
	`Facts=1`
	to enable output of the fact box. Enter
	`Facts=0`
	to disable output of the fact box. When using Facts as the only option, the tag will output only the fact box and no other parts of the story.

Header	Controls the output of the story header. The default is
	`Header=1.`
	Enter
	`Header=1`
	to enable output of the header. Enter
	`Header=0`
	to disable output of the header.

HotLinks	Enables the hotlink feature on the article text. The default is
	`HotLinks=0.`
	For this to work, a hotlinks.txt file must be in the root folder of the web site. The hotlinks.txt file is mainly used to change words in the body text to HTML links.
	`Google=http://www.google.com`
	`Aalborg=/section/Aalborg`

Karakter	Outputs the story's rating. The default is
	`Karakter=1.`
	Enter
	`Karakter=1`
	to enable output of the rating. Enter
	`Karakter=0`
	to disable output of the rating.

MediaOrder	This setting determines how the connected media (assets) are ordered `MediaOrder=(0|1|2)` The value specifies a sorting order: • 0 = Embeddable assets first, then direct linked assets first, then oldest assets first (default) • 1 = Embeddable assets first, then direct linked assets first, then newest assets first • 2 = Newest assets first, then embeddable assets first, then direct linked assets first

Modified	Outputs the story's last modification date. This option will use Article Modified.pbo when you use the option
	
	`useobject=0.`
	
	The default is for the modification date to be set in the category register. Enter
	
	`Modified=1`
	
	to enable output of the modification date. Enter
	
	`Modified=0`
	
	to disable output of the modification date.

OnlyShowChildConnections **From version 7.3.0sr3**	Used to control if the connections variable will only show child connections. `OnlyShowChildConnections="1"` The default behavior is to show both parent and child connections to the story. The edit interface will only show child connections, so enabling this feature on a tag will show the same connections as the edit interface.

Published	Outputs the story's publication date. This option will use ArticlePublished.pbo when you use the option
	`useobject=0.`
	The default is
	`Published=1.`
	Enter
	`Published=1`
	to enable output of the publication date. Enter
	`Published=0`
	to disable output of the publication date.

Summary	Controls the output of the story summary. The summary is shown in the summary field on the story. The default is
	`Summary=1.`
	Enter
	`Summary=1`
	to enable output of the summary. Enter
	`Summary=0`
	to disable output of the summary.

Tip	Controls the output of the basic Tip a Friend function. The function is controlled by the category setup, here you can override the category setting. The default is to enable output of the tip box in the category register.It is also possible to use the *article-tiplink *function to open a Tip a Friend window. Enter
	`Tip=1`
	to enable output of Tip a Friend. Requires that the option is enabled for the category.Enter
	`Tip=0`
	to disable output of Tip a Friend. When using Tip as the only option, only the tip box will be shown.

Text	Controls the output of the body text. The default is
	`Text=1.`
	Enter
	`Text=1`
	to enable output of the body text. Enter
	`Text=0`
	to disable output of the body text.

Title	Controls the output of the title. The default is
	`Title=1.`
	Enter
	`Title=1`
	to enable output of the title. Enter
	`Title=0`
	to disable output of the title.

Tittel2	Controls the output of the subtitle. The default is
	`Tittel2=1.`
	Enter
	`Tittel=1`
	to enable output of the subtitle. Enter
	`Tittel=0`
	to disable output of the subtitle.

Image options

Bilde	This option is the Norwegian version of the **Image** option.

Extrapictures	Controls output of the extra pictures imported on the story. You can also use a Photo Gallery. Please see Article linked photo gallery for information on Photo Galleries. The default is
	`Extrapictures=0.`
	Enter
	`Extrapictures=1`
	to enable output of extra pictures. Enter
	`Extrapictures=0`
	to disable output of extra pictures.

Image	Controls the output of paragraph photos. The default is
	`Image=1.`
	Enter
	`Image=1`
	to enables output of paragraph photos. Enter
	`Image=0`
	to disable output of paragraph photos. Only the main photo will be shown.

Paragraphpicturewidth	Sets the maximum width of paragraph pictures. This value is used by the image dll (**pbcsi.dll**) to resample high-resolution pictures uploaded to the system. The default is `Paragraphpicturewidth=0.` You can use any value between 1 and 999. If you set this value higher than the the width of the pictures you upload to the system, the system will not optimize the picture.

Picturealign	Controls the alignment of the main picture. The default is
	`Picturealign=Left.`
	Enter
	`Picturealign=Left`
	if the main pictures should be aligned on the left. Enter
	`Picturealign=Right`
	if the main pictures should be aligned on the right.

Picturebgcolor	Sets the background color for the picture and caption. The default is for no background color to be set.Any hex color triplet from #000000 to #FFFFFF. The value will be used as the background color for the table that holds the picture and the caption.

Pictureheight	Sets the maximum height of the main picture. This value is used by the image dll (**pbcsi.dll**) to resample high-resolution pictures uploaded to the system. The default is `Pictureheight=0.` You can use any value between 1 and 999. If you set this value higher than the the height of the pictures you upload to the system, the system will not optimize the picture.
Picturepadding	Controls how much padding the picture and caption have. This option is used by the cell padding parameter on the table. The default is `Picturepadding=1.` You can use any value between 0 to 99.
Picturewidth	Sets the maximum width of the main picture. This value is used by the image dll (**pbcsi.dll**) to resample high-resolution pictures uploaded to the system. The default is `Picturewidth=0.` You can use any value between 1 and 999. If you set this value higher than the the width of the pictures you upload to the system, the system will not optimize the picture.

Formatting options

These options control the layout of the story if object files are not used.

Faktawidth	Sets the width of the fact box in pixels. The default is `Faktawidth=120.` You can use any value between 10 and 1000.

Titlefont	Sets the font used for the story title. The string used here is used by the
	`<font face=ÓÓ>`
	tag output before the title. By default, the story inherits the document font.
	`Titlefont`
	can be set to any font or combinations of fonts.

Titlesize	This function sets the size of the story title. The legal values are 1-6. Setting the title size will result in the system outputting a `<font size=#>` where # is the value set.

Special options

adinsertspot	The ad insert function allows you to display an ad inserted into the story text. The function allows you to display any ad type in the story. The following code must be present in your article tag: `<pbs:article adinsertspot=SIDEAD02>` To show the ad, you must use AdInsert in your Article.pbo.

Gifoutput	Makes the system output the story as a GIF image instead of as text. This is used for making a preview for special access/pay sections of the site. This function creates a heavy system load and should be used sparingly. The default is
	
	`Gifoutput=0.`
	
	Entering
	
	`Gifoutput=1`
	
	enables output of the story in GIF format. Entering
	
	`Gifoutput=0`
	
	disables output of the story in GIF format.

Inherit	This option makes the article tag inherit the information about which story to show from the page that linked to it.
List	Makes the story output bullet points for each paragraph. The default is disabled. Enter `list="LIST"` in the tag to enable bullet points.
~~WAP~~ Deprecated	By entering `wap="WAP"` This will limit the length of the article page so it can be displayed by WAP devices. WAP stories behave like web stories. For example, images can be shown in both kinds of stories. However, large images do not work well with WAP.

- Article.pbo
    - ArticlePhotoGalleryThumbnailItems.pbo
- ArticleAccessControls.pbo
- ArticleAccessControlsItem.pbo
- ArticleByline.pbo
- ArticleConnectionItem.pbo
- ArticleConnections.pbo
- ArticleExtLogin.pbo
- ArticleExtraPictureItem.pbo
- ArticleExtraPictures.pbo
- ArticleFacts.pbo
- ArticleForumConnectionItem.pbo
- ArticleForumConnections.pbo
- ArticleGalleryConnectionItem.pbo
- ArticleGalleryConnections.pbo
- ArticleGeoDataItem.pbo
- ArticleGeoDataItems.pbo
- ArticleHead.pbo
- ArticleHotLinks.pbo
- ArticleLinkItem.pbo
- ArticleLinks.pbo
- ArticleLocation.pbo
- ArticleModified.pbo
- ArticleMultiMedia.pbo
- ArticleMultiMediaExtension.pbo
- ArticleMultimediaLinkItem.pbo
- ArticleMultiMediaList.pbo
- ArticleMultiMediaListItem.pbo
- ArticleMultimediaListItemExtension.pbo
- ArticleMultimediaListItemXXX.pbo
- ArticleMultiMediaPlaylistItem.pbo
- ArticleParagraph.pbo
- ArticlePhotoGalleryThumbnailItem.pbo
- ArticlePhotoGalleryThumbnails.pbo
- ArticlePicture.pbo
- ArticlePublished.pbo
- ArticleSummary.pbo
- ArticleTaxonomy.pbo
- ArticleTaxonomyItem.pbo
- ArticleTaxonomyItemPathItem.pbo
- InvalidArticle.pbo
- Tips.pbo and articles

Article.pbo

This is the main object file for the story. It supports the following variables:

<%AccessControlIdentifier%>	Contains the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.
<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of ArticleAccess Controls.pbo
AdInsert	The ad insert function allows you to display an ad inserted into the story text. The function allows you to display any ad type in the story. This function requires a special setting in the Publicus.ini. Please see Adinsert for more information.
<%ArtNo%>	Contains the story number.

<%ArticleStatus%>	Used to show different text based on the story status. This example shows how you can use the article status field to show different messages: `[%{ArticleStatus=0}` `  This story is not yet online.` `%]` `[%{ArticleStatus=1}` `  This story has been unpublished and is no longer available.` `%]` `[%{ArticleStatus=7}` `  This story is not available, because we do not have the rights to publish it online.` `%]` `[%{ArticleStatus=8}` `  This story has been archived and is only available from our paid archive.` `   ` `  <a href="<%inheritedlink%>&Subscribertype=ARCHIVED&Sectioncat=PROTECTED">Click here to log in to the archive</a> ` `%]` `[%{ArticleStatus=9|!Articlestatus}` `  The story goes here.....` `%]`
<%AssociatedKeywordLinks%>	Outputs external links that have the same keywords as the story. The following options are supported in the field: Count, Main category, Subcategory and Object class. This field has an attribute to specify which subcategory you would like to show. Example: `<%AssociatedKeywordLinks MainCategory=177987 SubCategory=10005 Count=5 ObjectClass=30%>`
<%AssociatedThemeLinks%>	Outputs external links that have the same theme as the story. The following options are supported in the field: Count, Main category and Object class.
<%AvgRanking%>	Contains the average ranking of a story.
<%Byline%>	Outputs the byline. It uses the file ArticleByline.pbo.
<%Category%>	Contains the category ID.
<%CategoryName%>	Contains the category name.

<%ConnectedEmbedMedia%>	Used to embed multimedia in the story. This field uses ArticleMultiMediaExtension.pbo.
	
	`[Extension]`
	
	is replaced by the media type. It is written:
	
	`<%ConnectedEmbedMedia%>`

<%ConnectedMedia%>	Lists connected multimedia objects. This field uses the object file ArticleMultiMediaList.pbo. It is written:
	
	`<%ConnectedMedia%>`
	
	For example,
	
	`<%ConnectedMedia%>`
	
	in **article50.pbo** searches for pbo files in the following order:
	
	`articleMultiMediaListItemPDF50.pbo`
	`articleMultiMediaListItemPDF.pbo`
	`articleMultiMediaListItem50.pbo`
	`articleMultiMediaListItem.pbo`

<%ConnectionItems%>	Outputs the links for connected stories, forums and galleries. It uses the files ArticleConnections.pbo and ArticleConnectionItem.pbo. Add following, to the <%ConnectionItems ShowRef=250%> to ensure that the leadin variable will contain summary text.
<%CurrentFlowStatusID%>	Contains the current status of the story within a flow using an ID. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%>	Contains the current status of the story within a flow using text. This variable is used in conjunction with Access Control.

<%DesignClass%>	The story's class.
	
	`[%{DesignClass=a}<h2>Breaking News</h2>%]`
	`[%{DesignClass=b}<b>Normal article</b>%]`
	
	Documentation may be found in the ARTSUBCLASS section.

<%ExtraFieldType%>	Contains the chosen type for the article. Types are defined in the variable EXTRAFIELDDEF section in the Publicus.ini.
<%ExtraPictures%>	Extracts extra pictures attached to a story. These pictures must be imported from your editorial system. To use this variable you must have the object files ArticleExtraPictures.pbo and ArticleExtraPictureItem.pbo.

<%Facts%>	Outputs the main fact box for the story. It uses the file ArticleFacts.pbo. ⚠ To show factbox items on paragraphs, you need to add <%Facts%> to ArticleParagraph<Your Objectclass>.pbo or ArticleParagraph.pbo.
<%Field_xxx%>	Extracts extra fields from the story. Replace `xxx` with the name of the field.
<%ForumConnectionItems%>	Outputs the links for connected forums. It uses ArticleForumConnections.pbo and ArticleForumConnectionItem.pbo. It is written: `<%ForumConnectionItems%>`
<%GalleryConnectionItems%>	Outputs the links to the connected galleries. It uses the files ArticleGalleryConnections.pbo and ArticleGalleryConnectionItem.pbo.
<%GeoDataItems%>	Output the geo data (locations on a map) attached to the story. For more information please see ArticleGeoDataItems.pbo and ArticleGeoDataItem.pbo.
<%GUID%>	Outputs the GUID of the article.
<%Head%>	Outputs the story head. It uses the ArticleHead.pbo file.
<%Inheritedlink%>	Contains the link to the page. It can be used for building a link to a printer friendly page like this: `<a href="<%inheritedlink%>&template=printart">Printer-friendly</a>`
<%IsoDate%>	Contains the stories date, in ISO format.YYYYMMDD
<%IsoModifiedDate%>	The date the story was last modified in ISO format. It is written: `<%IsoModifiedDate%>`
<%ISOPublishedDate%>	The publication date for the story in ISO format. It is written: `<%ISOPublishedDate%>` If publish control are activated and this publish control have a newer date then published date, will the value of publish control take over.

<%KeywordLinks%>	Outputs links to other stories that have the same keywords as the story. You set the count option like this:
	`<%KeyWordLinks count=10%>`
	To control the number of months the tag looks back, you use the count option like this:
	`<%KeywordLinks months=5%>`
	For more information please see pbs keywordlinks.

<%location%>	Used to output the location/dateline field. It uses ArticleLocation.pbo.
<%Keywords%>	Contains the keywords attached to the story.
<%Links%>	Outputs the external links. It uses the files ArticleLinks.pbo and ArticleLinkItem.pbo. You can also show links that have the same keywords or theme as the story.
<%MainPicture%>	Outputs the main story picture. It uses the file ArticlePicture.pbo.
<%ModifiedDate%>	Outputs the latest modification date for the story. This field does not use ArticleModified.pbo. You can use the field like this: `Published: <%PublishedDate%>[%{ISOModifiedDate > #ISOPublishedDate#} - Last modified: <%ModifiedDate%>%] .`
<%NewsGroupAutolink%>	**Deprecated**
<%OriginalStoryID%> From version 6.6.0sr47, 6.8.0sr19, 6.9.0sr1	Outputs the original story ID given to the story when imported.
<%ParagraphItems%>	Outputs all of the paragraphs for the story. It uses the file ArticleParagraph.pbo.
<%PublishedDate%>	The publication date for the story. The `<%PublishedDate%>` field does not use ArticlePublished.pbo.
<%RankingLink%>	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL **<%RankingLink%>** + '&rating=' + y Where y is the ranking given by a user.
<%Rating%>	Outputs the rating.
<%SEOLabel%> **From version 7.0.0sr11**	Outputs the SEO label of the story.
<%SingleSaleID%>	Contains the single sale ID associated to a story. It is possible to associate a single sale id to a story in the editing interface. This is used for Access Control.
<%Site%>	Contains the site code.

<%Summary%>	Outputs the summary for the story. It uses the file ArticleSummary.pbo.
<%TaxonomyList%>	Outputs the taxonomy words attached to the story using ArticleTaxonomy.pbo, ArticleTaxonomyItem.pbo, and ArticleTaxonomyItemPathItem.pbo. Specifying a delimiter will affect the output of the <%path%> variable in ArticleTaxonomyItem.pbo: `<%taxonomylist delimiter= -->%>`
<%Theme%>	Contains the story theme.

<%ThemeLinks%>	Outputs the links to other stories that have the same theme. To control the number of stories listed, use the count option like this:
	
	`<%ThemeLinks count=5%>`
	
	To control the number of months the tag looks back, use the count option like this:
	
	`<%ThemeLinks months=5%>`
	
	Please see pbs themelinks for more information.

<%ThumbnailGallery%>	Displays the first photo gallery attached to the story as thumbnails. This tag supports the option for setting columns, rows and object class like this: `<%ThumbnailGallery columns=4 rows=1 objectclass=1%>` Default value for both Columns and rows is 3. Note that when using parameters with this variable it is not possible to execute pbo modifiers. This means that if there are more than 9 images in the photo gallery this variable will only show the first 9 images if user need to execute pbo modifiers. The field uses the files ArticlePhotoGalleryThumbnails.pbo and ArticlePhotoGalleryThumbnailItems.pbo. These files have the same variables as the non-story versions documented in pbs photogallery-thumbnails
<%ThumbnailGalleryFull%>	This variable is identical to <%ThumbnailGallery%> except this variable will set the default value rows to 0 which means that all images in the photo gallery will be returned when user wants to execute pbo modifiers.
<%TipLink%>	Contains the link to the tip form. For more information please see Tip Functions.
<%Tips%>	Outputs the integrated tip form. For more information please see Tip Functions.
<%TotalNoofVotes%>	Contains the total number of votes on the story.
<%UsercommentsStatus%>	Outputs value 1 or 3 . value 1 means that usercomments are activated on the article, value 3 means that usercomments are locked on the article. If the variable is empty, this means that usercomments are not activated on the article

<%UserCommentsLink%>	Contains the link to the submit form. We recommend that this request always is inside a ajax call, to make sure that the output not will be a part of the article cache. `$('#comments').load('<%UserCommentsLink%>');`
<%UserComments%>	Outputs comments on article. For more information please see User Comments.

ArticlePhotoGalleryThumbnailItems.pbo

ArticleAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of ArticleAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

ArticleAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

ArticleByline.pbo

The byline file operates in different modes depending on category settings, whether or not the person is found in the staff register, and whether or not Smart Bylines are enabled. To use the Smart Bylines feature, you must use `ArticleByline.pbo`.

If the Smart Bylines feature is enabled in the Publicus.ini, the byline will show the first name, surname, and e-mail address of the staff member (as entered in the Staff component) selected in the user interface.

```
UseSmartByline=1
```

If Smart Bylines is not used, a byline is only processed if it is active and in the category register. If the byline is set to active, you can use fields from the staff register to format the byline. This can be overridden by entering

```
AlwaysUseBylineObject=1
```

in the Publicus.ini. Alternatively, you can make the system output the original byline written in the story.

Templates example

Here is an example on supporting Smart Byline and regular byline:

```
[%{Counter}
  [%{Counter>1}[%{TotalCount=#Counter#}and%|%,%]%|%Written by%]
  [%{Counter=1}<a href="/section/staff&staffID=<%StaffID%>"><%BylineNamePart%></a>%]
  [%{Counter=2}<a href="/section/staff&staffID=<%StaffID2%>"><%BylineNamePart2%></a>%]
  [%{Counter=3}<a href="/section/staff&staffID=<%StaffID3%>"><%BylineNamePart3%></a>%]
  [%{Counter=4}<a href="/section/staff&staffID=<%StaffID4%>"><%BylineNamePart4%></a>%]
  [%{Counter=5}<a href="/section/staff&staffID=<%StaffID5%>"><%BylineNamePart5%></a>%]
%|%
  Written by <%Byline%>
%]
```

When not using smart bylines, the Online Robot will try to find staff information based on first name and last name in the byline field. This requires removing noise words from the byline field before processing. Noise words that must be removed from the byline field are defined in OnlineRobot_StoryExport.ini.

```
[[Sitecode_SITE_]SETTINGS]
NoiseWordsInByline="BY "," and ","Staff Writer", "#13#10"
```

Online starts from left to right and combines 2 words (firstname + ' ' + lastname) at a time before looking up the staff.

<%Address1%>	The first part of the address as defined in the staff registry.
<%Address2%>	The second part of the address as defined in the staff registry.
<%Address3%>	The third part of the address as defined in the staff registry.
<%Address4%>	The fourth part of the address as defined in the staff registry.
<%Byline%>	Outputs the names of all authors with the names as links to the e-mail addresses of those authors the system finds in the register. The function also outputs the other lines in the story.
<%BylineEmailPart%>	Contains the author's e-mail address as saved in the staff register. The same functions as for **<%BylineNamePart%>** are available here.
<%BylineNamePart%>	Contains the author's name as registered in the staff register. If you have two names in the byline you also have the field **<%BylineNamePart2%>**. This function supports up to 9 names in the byline.
<%City%>	Used for Smart Bylines. Contains the author's city as saved in the staff register.
<%CreatedByUserId%>	Contains the creator's user ID if the story is a blog entry.
<%CreatedByUserFirstName%>	Contains the creator's first name if the story is a blog entry.
<%CreatedByUserSurName%>	Contains the creator's last name if the story is a blog entry.
<%CreatedByUserEmailAdr%>	Contains the creator's email address if the story a blog entry.
<%CreatedByUserNickName%>	Contains the creator's nickname if the story a blog entry.
<%Counter%>	Used for Smart Bylines. Contains the current author number.
<%Department%>	Used for Smart Bylines. Contains the author's department as saved in the staff register.
<%DesignClass%>	Contains the design class used.
<%Extension%>	Used for Smart Bylines. Contains the author's work phone number extension as saved in the staff register.

<%Fax%>	Used for Smart Bylines. Contains the author's fax number as saved in the staff register.
<%Field_xxx%> From version 6.9.0sr2	Used for Smart Bylines. Extracts extra fields for the staff member. Replace **xxx** with the name of the field. E.g., `<%Field_Facebook%>`
<%Firstname%>	Used for Smart Bylines. Contains the author's first name as saved in the staff register.
<%Fylkenr%>	Used for Smart Bylines. Contains the author's fylkenr as saved in the staff register.
<%Group%>	Used for Smart Bylines. Contains the author's group as saved in the staff register.
<%ImgLink%>	Contains the URL of the author's picture. For additional authors use the field **<%ImgLink2%>**. This tag supports up to 9 people.
<%Information%>	Used for Smart Bylines. Contains the author's information as saved in the staff register.
<%IsPrimary%>	Used for Smart Bylines. `IsPrimary=1` indicates that the author is the primary author.
<%Kommunenr%>	Used for Smart Bylines. Contains the author's kommunenr as saved in the staff register.
<%LastName%>	Used for smart bylines. Contains the author's surname as saved in the staff register.
<%Line1%>	Contains the first line of the byline field.
<%Line2%> etc...	You can increase the line number to output the different lines of the byline field.
<%MobilePhone%>	Used for Smart Bylines. Contains the author's mobile phone number as saved in the staff register.
<%Name%>	Outputs the names of all authors without e-mail links. The function also outputs the other lines in the story.
<%Namecount%> **(Deprecated)** **Use <%Counter%> instead.**	Contains the number of authors the system has recognized. It can be used like this: `[%{namecount=1}<%BylineNamePart1%>%|%<%BylineNamePart%> and <%BylineNamePart2%>%]`
<%Office%>	Used for Smart Bylines. Contains the author's office as saved in the staff register.
<%PagerNumber%>	Used for Smart Bylines. Contains the author's pager number as saved in the staff register.
<%Position%>	Used for Smart Bylines. Contains the author's position as saved in the staff register.

<%PrivateEmailAddress%>	Used for Smart Bylines. Contains the author's private email address as saved in the staff register.
<%PrivatePhone%>	Used for Smart Bylines. Contains the author's private phone number as saved in the staff register.
<%PrivateWebAdress%>	Used for Smart Bylines. Contains the author's private web address as saved in the staff register.
<%StaffID%>	Contains the value from the staff ID field. This allows you to link to the information page for one author. The field is used like this: `<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>"><%BylineName Part%></a>` Please see pbs staffinfo for more information.
<%Surname%>	Used for Smart Bylines. Contains the author's surname as saved in the staff register.
<%TotalCount%>	Used for Smart Bylines. Contains the total count of authors on a story.
<%WorkEmailAddress%>	Used for Smart Bylines. Contains the author's work email address as saved in the staff register.
<%WorkPhone%>	Used for Smart Bylines. Contains the author's work phone number as saved in the staff register.
<%WorkWebAddress%>	Used for Smart Bylines. Contains the author's work web address as saved in the staff register.
<%Zipcode%>	Used for Smart Bylines. Contains the author's zip code as saved in the staff register.

ArticleConnectionItem.pbo

<%Count%>	The total item count.
<%Counter%>	The item count starting from 1.
<%Date%>	The publication date for the story.
<%Heading%>	The heading for the linked story.
<%IsLastItem%>	Will be '1' for the last item, else blank.
<%ImageLink%>	The link to the main photo of the connected story. You use the field like this: `[%{ImageLink}<img src="<%ImageLink%>?MaxW=120&MaxH=80" border=0 hspace=1 vspace=1 alt=" " align="right">%]`
<%Leadin%>	Outputs the summary if the leadin is larger than 0. To ensure that the leadin is larger than 0, use ShowRef.
<%Link%>	The address of the connected story. From version 7.1.0sr16, 7.2.0sr9: If there are a SEO label associated with the story and the feature otherwise is enabled, the link will be SEO friendly.

<%SEOLabel%> **From version 7.1.0sr16, 7.2.0sr9**	The SEO label of the story.
<%Status%>	The status for the linked story.
<%Source%>	The Source for the linked story.
<%PayStatus%>	The PayStatus for the linked story.
<%TeaserHeading%>	The teaser heading of the connected story.
<%TeaserImageLink%>	The teaser image Link of the connected story. You use the field like this: `[%{TeaserImagelink}<img src="<%TeaserImagelink%>?MaxW=120&MaxH=80" border=0 hspace=1 vspace=1 alt=" " align="right">%]`
<%TeaserSummary%>	The teaser summary of the connected story.
<%TeaserUrl%>	The External URL of a teaser link, if available

ArticleConnections.pbo

<%ConnectionItems%>	Outputs the story items.
<%ForumConnectionItems%>	Outputs the forum items.
<%GalleryConnectionItems%>	Outputs the photo gallery items.

ArticleExtLogin.pbo

<%AccessControlIdentifier%>	Indicates the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusID%>	Represents the current status of the story within a flow using an ID. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%>	Represents the current status of the story within a flow using text. This variable is used in conjunction with Access Control.
<%SingleSaleID%>	Represents the single Sale ID associated to a story. It is possible to associate a single sale id to a story in the editing interface. This is used for Access Control.
<%StandardText%>	This field contains the text to output. For example, *You must be a subscriber to access this story. To buy a subscription...*

This file gets all of the same variables as Article.pbo (in addition to the ones mentioned above).

ArticleExtraPictureItem.pbo

<%ArtNo%>	Contains the image ID number.
<%Category%>	The image category.
<%Counter%>	Contains the picture counter. It starts on 1 for the first picture.
<%DesignClass%>	Contains the design class used.
<%Img%>	HTML image tag to make sure the image is displayed.

<%ImgLink%>	The link to the image.
<%IsLastItem%>	This field has the value 1 on the last picture.
<%ISODate%>	The date of the image.
<%Site%>	The two letter site code.
<%Caption%>	The caption for the picture.
<%CaptionTitle%>	The title for the picture.
<%Photo%>	The photo byline.
<%ImageHeight%> **From version 7.2.0sr24**	The height of the image in pixels.
<%ImageWidth%> **From version 7.2.0sr24**	The width of the image in pixels.
<%ImageOrientation%> **From version 7.2.0sr24**	The orientation of the image. Value is "L" or "P" - for landscape and portrait, respectively.

All fields from ArticlePicture.pbo are also supported.

ArticleExtraPictures.pbo

<%Count%>	Shows the number of pictures.
<%DesignClass%>	Contains the design class used.
<%Items%>	Outputs the items.
<%OriginalStoryID%> **From version 6.6.0sr47, 6.8.0sr19, 6.9.0sr1**	Outputs the original story id given to the story when imported.

ArticleFacts.pbo

Used to define the fact box. It is used for all fact boxes in the story. The file `ArticleFacts_0.pbo` will only be used for the main fact box. `ArticleFacts_1.pbo` will be used for the fact box in the first paragraph, `ArticleFacts_2.pbo` for the second paragraph, and so on. `ArticleFacts.pbo` is used for all fact boxes that do not have their own template.

<%Align%>	Outputs the alignment of the fact box. In the editing interface, users can select either left or right.
<%DesignClass%>	Contains the design class used.
<%Heading%>	Outputs the title for the fact box.
<%Text%>	Outputs the text for the fact box.
<%Width%>	Outputs the value of the FaktaWidth option in the article tag, which is used for controlling the width of the fact box.
<%ImageHeading%>	Outputs the title for the image. Only available for the main fact box.
<%ImageBody%>	Outputs the caption for the image. Only available for the main fact box.
<%ImageByline%>	Outputs the credit for the image. Only available for the main fact box.
<%AltText%>	Outputs AltText for the image. Only available for the main fact box.
<%SeoFilename%>	Outputs seo filename for the image. Only available for the main fact box.
<%ImageSrc%>	Outputs source for the image. Only available for the main fact box.
<%ImageHeight%> **From version 7.2.0sr24**	The height of the image in pixels. Only available for the main fact box.

<%ImageWidth%> **From version 7.2.0sr24**	The width of the image in pixels. Only available for the main fact box.
<%ImageOrientation%> **From version 7.2.0sr24**	The orientation of the image. Value is "L" or "P" - for landscape and portrait, respectively. Only available for the main fact box.

ArticleForumConnectionItem.pbo

This file is used by **<%ForumConnectionItems%>** in both Article.pbo and ArticleConnections.pbo.

<%\Date%>	The creation date for the forum.
<%Link%>	The address of the connected forum.
<%ObjectClass%>	If you need to use the current object class with the forum template, add it to the link like this: `<%Link%>&ObjectClass=<%ObjectClass%>`
<%\Title%>	The heading of the linked forum.

ArticleForumConnections.pbo

This file is used with **<%ForumConnectionItems%>** in Article.pbo.

<%ConnectionItems%>	Outputs the forum items.

ArticleGalleryConnectionItem.pbo

<%ImgLink%>	Contains the link to the first image in the gallery.
<%Link%>	Contains the link to the gallery.
<%Title%>	Contains the title of the gallery.

ArticleGalleryConnections.pbo

Used when a gallery is attached to a story.

<%ConnectionItems%>	Shows all gallery all items.

ArticleGeoDataItem.pbo

<%Name%>	Contains the name of the geo data item, as for example "The summer house"
<%Location%>	The address or other text representation of the location, as for example "Petersbuger Str. 91, Berlin"
<%Latitude%>	The latitude of the location, if known. Latitude will be a number between -90 and 90. If unknown it is blank (not to be confused with 0 which is a valid, although rarely used, latitude.)
<%Longitude%>	The longitude of the location, if known. Longitude will be a number between -180 and 180. If unknown it is blank (not to be confused with 0 which is a valid, although rarely used, longitude.)
<%Counter%>	A Counter
<%Zoom%>	Contains the zoom level, a number from 1 - 19.

<%MapType%>	Contains the map type as specified in the Google API. Supports the following map types: normal, hybrid, satellite, and terrain.

ArticleGeoDataItems.pbo

<%ArticleGeoDataItems%>	Outputs all Geo data items attached to the story.
<%Items%>	This is the same as ArticleGeoDataItems.
<%Count%>	Number of Geo locations associated to a story

The individual Geo Data items are rendered using ArticleGeoDataItem.pbo.

ArticleHead.pbo

Controls output of the story header. It supports the following fields.

<%ArtNo%>	Contains the story number.
<%Category%>	Contains the category ID.
<%DesignClass%>	Contains the design class used.
<%Heading%>	Contains the story Title
<%Heading2%>	Contains the story SubTitle
<%Site%>	Contains the site code.

<%SuperHeading%>	Contains the story Kicker. You can use object scripts to force capitalization of the super heading. For example:
	`<%Superheading$u%>`
	will show as 'THIS IS A TEST'
	`<%Superheading$p%>`
	will show as 'This is a test'
	`<%Superheading$n%>`
	will show as 'This Is A Test'
	`<%Superheading$l%>`
	will show as 'this is a test'

ArticleHotLinks.pbo

This file is output hotlinks matching the hotlinks.txt file.

A standard example, changing the matches to links

```
<a href="<%Value%>"><%Match%></a>
```

<%Count%>	The number of times the hotlink word is matched in the body text

<%Counter%>	The number of the currently match in the body text.
<%Match%>	Contains the text matching the left side of the equal in hotlinks.txt.
<%Value%>	Contains the value of the right side of the equal in the hotlinks.txt class used.

ArticleLinkItem.pbo

This file is output for each of the external links in the story.

<%AgeRating%>	Contains the age rating.
<%Counter%>	The counter is the number for the link. The fist link is given the value 1 and the second link the value 2. This value is increased by 1 for each additional link.
<%Description%>	Contains the description of the site.
<%DesignClass%>	Contains the design class used.
<%ImageLink%>	Contains a link to the photo defined in Story External Links in the Edit interface.
<%Link%>	Contains the link to the external site.
<%LinkID%>	Contains the link's ID.
<%LinkMainCategory%>	Contains the main category of the linked story.
<%LinkMainCategoryID%>	Contains the main category ID of the linked story.
<%LinkRegion%>	Contains the region of the linked story.
<%LinkSubCategory%>	Contains the subcategory of the linked story.
<%LinkSubCategoryID%>	Contains the subcategory ID of the linked story.
<%LinkSubRegion%>	Contains the subregion of the linked story.
<%LogoLink%>	Contains a link to the logo defined in External Links in the edit interface.
<%Name%>	Contains the name of the external site.
<%QualityRating%>	Contains the quality rating.

ArticleLinks.pbo

Used when you have links to external sites in the story.

<%DesignClass%>	Contains the design class used.
<%LinkItems%>	This field outputs all the links.

ArticleLocation.pbo

Controls output of the dateline.

<%DesignClass%>	Contains the design class used.
<%Location%>	This field contains the dateline.

ArticleModified.pbo

This object file is only used when you use Useobject.

<%DesignClass%>	Contains the design class used.
<%ISOModifiedDate%>	Contains the publication date in ISO format.

<%ModifiedDate%>	The publication date.

ArticleMultiMedia.pbo

The fallback file used if you have not created an object file for the media type. Because different file types are managed differently, it is nearly impossible to write a general object file for all media types. This file should tell the user that the support file for the media format is not installed on the server.

<%DesignClass%>	Contains the design class used.

ArticleMultiMediaExtension.pbo

ArticleMultiMedia[Extension].pbo

Used for embedded multimedia objects.

```
[Extension]
```

is replaced by the media type extension. See the example section for the structure for different formats. This file is also used by the tag **<pbs:articlemultimedia>**

<%Align%>	Contains the align link from the story properties page.
<%AutoStart%>	Contains 0 or 1, depending on whether or not the file should auto start.
<%DirectDownload%>	Has the value 1 if the item is to be downloaded directly.
<%DesignClass%>	Contains the design class used.
<%DirectLink%>	Has the value 1 if the item has direct link active in the edit interface.
<%ExternalURL%>	Contains the external URL of the item.
<%FileDescription%>	A description of the object.
<%FileName%>	The filename of the object.
<%FileSize%>	The size of the object.
<%Height%>	Contains the height from the properties page for the media type.
<%Link%>	Contains the link to the item.
<%MediaDescription%>	The description of the media type.
<%MediaTypeExt%> **From versions 6.6.0sr47, 6.8.0sr19, 6.9.0sr1**	The media type extension.
<%OrgFileName%>	The original filename.
<%Playlist%>	Contains the playlist index of the media file. It is rendered through the ArticleMultiMediaPlaylistItem.pbo object file.
<%Start%>	The start field.
<%Stopp%>	The stop field.
<%StreamLink%>	Contains the stream link to the item.
<%Width%>	Contains the width from the properties page for the media type.

ArticleMultimediaLinkItem.pbo

Used for linking to movie files.

<%Description%>	Contains a description of the content.

<%DesignClass%>	Contains the design class used.
<%Format%>	Contains the name of the format.
<%MovieLink%>	Contains the link to the movie file.
<%Speed%>	Contains the speed value necessary to view the movie stream.

ArticleMultiMediaList.pbo

Lists the connected media items.

<%MediaItems%>	Lists all the connected items.

ArticleMultiMediaListItem.pbo

Used for all connected items.

<%AutoStart%>	Contains either 1 or 0. Controls whether or not embedded clips auto start.
<%Counter%>	Contains the counter value.
<%DesignClass%>	Contains the design class used for the article.
<%DirectDownload%>	Has the value 1 if the item is to be downloaded directly.
<%DirectLink%>	Contains the link for direct download.
<%Embed%>	Has a value if the file can be embedded. This is defined in media type setup.
<%ExternalURL%>	Contains the external URL if such exists for the mediafile.
<%FileDescription%>	The title of the file.
<%FileName%>	The internal file name.
<%FileSize%>	The file size in bytes.
<%Height%>	The height set in the media type.
<%Link%>	The link to the item.
<%MediaDescription%>	A description of the media type.
<%MediaTypeExt%>	Outputs the media type extension.
<%OrgFileName%>	The original file name.
<%PopupLink%>	Contains the link to the popup page. This page is used if the item is not to be downloaded directly.
<%Start%>	Controls the start point in the clip.
<%Stopp%>	Controls the end point in the clip.
<%StreamLink%>	The streaming link to the item.
<%Width%>	The width set in the media type.

ArticleMultimediaListItemExtension.pbo

ArticleMultimediaListItem[Extension].pbo

You can override the object file used for each media type. This is an optional feature for special cases.

ArticleMultimediaListItemXXX.pbo

Where XXX is your Castfire video asset type. Used for Castfire integration.

```
<embed
id="castfire_player_<%OrgFileName%>_flv"
class="castfire_player"
src="<%Link%>"
quality="high"
wmode="transparent"
name="castfire_player_<%OrgFileName%>_flv"
allowScriptAccess="always"
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer"
width="<%Width%>"
height="<%Height%>">
</embed>
```

ArticleMultiMediaPlaylistItem.pbo

Used to render the <%Playlist%> variable in ArticleMultiMediaExtension.pbo.

<%Counter%>	Countains the counter value.
<%DesignClass%>	Design class of the story.
<%Seconds%>	Runtime in seconds.
<%Time%>	Formatted time.
<%Title%>	The title of the media.

ArticleParagraph.pbo

<%AltText%>	Outputs the information from the image Alt text field.
<%Blockid%>	Contains the ID for the paragraph. The first paragraph is given the ID 1. The number is increased by one for each paragraph.
<%Body%>	Outputs the paragraph.
<%DesignClass%>	Contains the design class used.
<%Counter%>	Outputs the current paragraph count.
<%Facts%>	Outputs the fact box for the paragraph.
<%Heading%>	Outputs the heading for the paragraph.
<%Html%>	This field has the value 1 if the HTML flag is turned on for the paragraph. Otherwise, the field is blank.
<%IsLastItem%>	Is set if it is the last paragraph in the story.
<%Location%>	Outputs the dateline. To limit it to the first paragraph, you can use the following code: `[%{location&Counter=1}<%location%>%]` The publicus.ini setting Stedopt in section GENERELT must be set to 2 in order for this variable to be available in this object file.
<%Picture%>	Outputs the paragraph picture.Uses object: ArticlePicture.pbo
<%TotalCount%>	Outputs the total count of paragraphs in the story.

ArticlePhotoGalleryThumbnailItem.pbo

This file has the same variables as the non-story version documented in the chapter pbs photogallery-thumbnails.

ArticlePhotoGalleryThumbnails.pbo

This file has the same variables as the non-story version documented in the chapter pbs photogallery-thumbnails.

ArticlePicture.pbo

Used to display the pictures on the story. You can also use numbering in the file name to determine which pictures will use the file. The file ArticlePicture_0.pbo will only be used for the main picture. ArticlePicture_1.pbo will be used for the picture on the first paragraph, ArticlePicture_2.pbo for the second paragraph, and so on. ArticlePicture.pbo is used for all pictures that do not have their own template.

If you have integration with QPS, you might get pictures that contain only the caption and not the picture itself. These images have the size -1 by -1 pixels. You can use object script to check that the size is larger than 1 to avoid outputting just the caption.

<%Align%>	This field contains the value from the editing interface for aligning the picture left or right. For the main picture, it uses the value from the PictureAlign option in the article tag. It is usually used like this in the table that wraps the picture: `<table align="<%Align%>">`
<%AltText%>	Outputs the information from the image Alt text field.
<%Caption%>	The caption for the picture.
<%CaptionTitle%>	The title for the picture.
<%DesignClass%>	Contains the design class used. The DesignClass variable will always show the short name. The long name is only used for display purposes in the edit interface.
<%Height%>	Works like **<%width%>** for height.
<%ImgKeyParams%>	Contains the key parameters for the picture. These parameters are used when you want to use functions like zoom. It is usually used like this: `<a href="/apps/pbcs.dll/misc?url=/templates/zoom.pbs&<%ImgKeyParams%>">Zoom</a>` This function is described in the section pbs link and images.
<%ImgLink%>	Contains the URL that is used as source for the **<img>** tag. It is possible to use the options **Title=1** and **Logo=/somefile.gif** in conjunction with this field so that the tag can be used like this: `<img src="<%ImgLink%><%MaxWidth%>&title=1&logo=/graphics/logo.gif">`
<%MainPicture%>	This field has the value 1 if the picture is the main picture.

<%MaxHeight%>	Contains the value from the options **PictureHeight** or **ParagraphPictureHeight**. It outputs
	`&MaxH=XXX`
	where
	`XXX`
	is the value from the option. It is used with the **<%ImgLink%>** field.

<%MaxWidth%>	Contains the value from the options PictureWidth or ParagraphPictureWidth. It outputs
	`&MaxW=XXX`
	where
	`XXX`
	is the value from the option. It is used with the **<%ImgLink%>** field.

<%Orientation%>	This field contains the value
	`L`
	if the image orientation is landscape and
	`P`
	if it is portrait.

<%OriginalStoryID%> **From version 6.6.0sr47, 6.8.0sr19, 6.9.0sr1**	Outputs the original story id given to the story when imported.
<%Photo%>	The photo byline.

<%Width%>	Contains the width of the picture. **<%width%>** outputs the value passed in from
	
	`<pbs:article picturewidth=XXX>`
	
	if
	
	`XXX`
	
	is defined and is less than the image's actual width. If the image's width is larger, it outputs the actual width.It can be used with the **<%ImgLink%>** field or in the table tag like this:
	
	`<table width="<%Width#+8%>">`
	
	When you use the option like this, you will get spacing around the picture table.

ArticlePublished.pbo

This object file is only used when you use Useobject.

<%DesignClass%>	Contains the design class used.
<%ISOPublishedDate%>	Contains the publication date in ISO format.
<%PublishedDate%>	Contains the publication date.

ArticleSummary.pbo

Controls output of the story summary.

<%DesignClass%>	Contains the design class used.
<%Location%>	Outputs the content of the location field.
<%Summary%>	Outputs the content of the summary field.

ArticleTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.

ArticleTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
~~**<%url%>**~~ **obsolete**	~~Outputs the URL associated with the taxonomy word. This variable is no longer in general use and will always be empty.~~
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.

<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

ArticleTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

InvalidArticle.pbo

If a web guest links to a story that no longer exists, she will get see error message: *Invalid article key.* You can override this so that the web guest will see a more understandable error message.

<%Message%>	The message from the system.
<%StoryCategory%>	The category in which the story is placed.
<%StoryDate%>	The story's date.
<%StoryID%>	The story's ID.
<%StorySite%>	The site on which the story is published.

Tips.pbo and articles

For information about the integrated tip form, see the section pbs article-tiplink.

pbs articledate

<pbs:articledate>

This tag outputs the date the story was published.

Options

Dateformat	This option's parameters are described in Formatting dates.

pbs articlefield

<pbs:articlefield>

If your story interface is set up with extra fields like a quote or a slug, it is possible to extract these fields on the article template. Which fields you have available will vary depending on your site setup.

This tag can be used on all article templates.

Options

Field	The field option is used for extracting the field. It is used like this: `<pbs:articlefield field="xxx">` Where xxx is the name of the field. Extra fields can also be shown in the article object files using Field_XXX.

pbs article-links

<pbs:article-links>

This tag is used to show stories that have been liked to the story being viewed. These stories are linked manually, through the web-based user interface. You can also use ConnectionItems in the Article.pbo.

Options

Forwardonly	This option will make the tag link only to the linked stories, and not from the linked story to the main story. Subarticles ignores this option.
Includegallery	This option will make the tag output links to photo galleries.

Object files

The tag uses ArticleConnections.pbo.

pbs articlemultimedia

<pbs:articlemultimedia>

This tag is used to output the code needed to show streamed media embedded in a pop-up window. This function is used when you turn off direct download on a linked asset.

This function uses the template **ArticleMultiMediaPopup.pbs**. The popup link in the object file ArticleMultiMediaListItem.pbo links to it.

This tag uses ArticleMultiMediaExtension.pbo, which is also used for the embedded streamed media.

This tag does not have any options.

pbs artikkel-urls

<pbs:artikkel-urls>

This tag outputs the same information as Links in the Article.pbo. You can also show links that have been tagged with the same keywords or theme using the links tag: `<pbs:links usetheme=1>`. The tag is documented here: pbs links.

Object files

The object file is described in the section ArticleLinks.pbo.

pbs keywordlinks

<pbs:keywordlinks>

Options

Antall	This option works the same way as **Count** on the Sectionfront tag. Please see Count
Visref	This option works the same way as **Leadin** on the Article tag. Please see Leadin.

- KeywordLinkItem.pbo
- KeywordLinks.pbo

KeywordLinkItem.pbo

The tag will use this object file if it exists. This file is used to build each item in the listing.

<%Counter%>	Countains the counter value.
<%Date%>	The date the story was published.
<%Link%>	The URL that links to the story.
<%Title%>	The title of the story.

KeywordLinks.pbo

The tag will use this object file if it exists.This file is used to build the framework around the stories.

<%Items%>	This is where the system will output the items.
<%MoreLink%>	This parameter will make links to all the stories with same keyword.

pbs photogallery-links

<pbs:photogallery-links>

Used to show Gallery links. This tag outputs the same information as GalleryConncetionItems in the Article.pbo.

Object file

The object file is described in the section GalleryConncetionItems.

pbs themelinks

<pbs:themelinks>

This tag is used for theme links, also called related stories. It is used to display links to other stories that have been tagged with the same theme. You can also use ThemeLinks in the Article.pbo. The tag for displaying all related stories is described in the section Pbsrelated.

Options

Antall	This option works the same way as **Count** on the Sectionfront tag. Please see Count.
Bgcolor	Sets the background color for the header. This is only used when object files are not used. The default is the HTML color #CCCCCC. You can use any HTML color.
Color	Sets the text color in the header. This is only used when object files are not used. The default is the default color for the text page.This option is set by putting `color="#??????"` inside the tag.
Bredde	Sets the width to be used the by tag. Only used when object files are not used.
***Fontsize ***	This option sets the font size. Only used when object files are not used. The default is `Fontsize=2.` You can use any size from 1 to 6.

Ledetekst	Sets the title for the listing. You can use any text. If you enter `ledektekst=""` it will not output any text.
Visref	This option works the same way as **Leadin** on the Article tag. Please see Leadin.

- ThemeLinkItem.pbo
- ThemeLinks.pbo

ThemeLinkItem.pbo

This file is used for building each item in the listing.

<%Counter%>	Countains the counter value.
<%Date%>	The date the story was published. It can be formatted with object script.
<%Link%>	This is the URL that links to the story.
<%Title%>	The title of the story.

ThemeLinks.pbo

This file is used for building the framework around the stories.

<%Items%>	This is where the system will output the items.
<%MoreLink%>	This parameter will make links to all the stories with same theme.

Other links

Linking to a keyword index

Use the following tag to link to the keyword index for a category:

```
<a href="<pbs:prog>/index?Category=<pbs:categoryid>">View the <pbs:category main>
keyword index</a>
```

Overview Tags

The tags in this chapter are used for building overviews of stories. This is similar to a section front and can be used as such. They can be used on most templates.

- pbs agent
  - Agent.pbo
  - AgentItem.pbo
- pbs agentcategory
- pbs contentlist
  - ContentList.pbo
  - ContentListByline.pbo
  - ContentListFacetItem.pbo
  - ContentListFacetList.pbo
  - ContentListFilterItem.pbo
  - ContentListFilterList.pbo
  - ContentListItem.pbo
  - ContentListItemAccessControls.pbo
  - ContentListItemAccessControlsItem.pbo
  - ContentListItemImageItem.pbo
  - ContentListItemTaxonomy.pbo
  - ContentListItemTaxonomyItem.pbo
  - ContentListItemTaxonomyItemPathItem.pbo
- pbs contentstats
  - ContentStats.pbo
  - ContentStatsItem.pbo
- pbs freestyle
  - FreestyleStoryArticleConnectionItem.pbo
  - FreestyleStoryItem.pbo
  - FreestyleStoryItemByline.pbo
  - FreestyleStoryItemConnectionItem.pbo
  - FreestyleStoryItemExternalLinkItem.pbo
  - FreestyleStoryItemExternalLinks.pbo
  - FreestyleStoryItemForumConnectionItem.pbo
  - FreestyleStoryItemGalleryConnectionItem.pbo
  - FreestyleStoryItemInternalLinks.pbo
  - FreestyleStoryItemKeywordLinkItem.pbo
  - FreestyleStoryItemKeywordLinks.pbo
  - FreestyleStoryItemThemeLinkItem.pbo
  - FreestyleStoryItemThemeLinks.pbo
  - FreestyleStoryList.pbo
- pbs history
  - History.pbo
  - HistoryItem.pbo
- pbs keywordindex
  - KeywordIndexHeader.pbo
  - KeywordIndexLink.pbo
  - KeywordIndexTitle.pbo
- pbs mostread
  - MostRead.pbo
  - MostReadItem.pbo
  - MostReadItemAccessControls.pbo
  - MostReadItemAccessControlsItem.pbo
  - MostReadItemTaxonomy.pbo
  - MostReadItemTaxonomyItem.pbo
  - MostReadItemTaxonomyItemPathItem.pbo
- pbs newslist
  - News.pbo
  - NewsItem.pbo
  - NewsItemAccessControls.pbo
  - NewsItemAccessControlsItem.pbo
  - NewsItemArticleConnectionItem.pbo
  - NewsItemByline.pbo
  - NewsItemConnectionItem.pbo
  - NewsItemExternalLinkItem.pbo
  - NewsItemExternalLinks.pbo
  - NewsItemExtraPictureItem.pbo
  - NewsItemExtraPictures.pbo
  - NewsItemForumConnectionItem.pbo
  - NewsItemForumGroupLink.pbo
  - NewsItemGalleryConnectionItem.pbo
  - NewsItemGeoData.pbo

        - NewsItemGeoDataItem.pbo
        - NewsItemInternalLinks.pbo
        - NewsItemKeywordLinkItem.pbo
        - NewsItemKeywordLinks.pbo
        - NewsItemMultiMediaExtension.pbo
        - NewsItemParagraphItem.pbo
        - NewsItemTaxonomy.pbo
        - NewsItemTaxonomyItem.pbo
        - NewsItemTaxonomyItemPathItem.pbo
        - NewsItemTaxonomyWordItem.pbo
        - NewsItemTaxonomyWords.pbo
        - NewsItemThemeLinkItem.pbo
        - NewsItemThemeLinks.pbo
        - RelStoriesItem.pbo
        - RelStoriesItems.pbo
        - URL Parameters and overview tags
    - pbs related
        - ThemeLinkItem.pbo and overview tags
        - ThemeLinks.pbo and pbs related
    - pbs subcategories
        - SubcatSimpleLink.pbo

pbs agent

<pbs:agent>

This tag is used for listing stories that come into the system from press agencies. The tag was previously used to list the most popular stories on a site, functionality that has been moved to the pbs mostread tag.

This is an example of what the tag could look like:

```
<pbs:agent leadin=0 count=5 category="ntbu">
```

This tag will use object files if they exist. Object files are **not** available when you use the **Channels** option in the Publicus.ini.

ArtKeys	Extracts stories from the press agency NTB. The option is used like this:
	
	`Artkeys="SOS,ARB,KUL,POL"`
	
	The possible keys are: `ARB, FRI, KRI, KRE, KUL, MED, NAT, KUR, POL, REL, SOS, SPO, ULY, UTD, VIT, VAR, OKO`.Filtering can be used on this tag to extract stories based on the value in a key. The option is then used like this:
	
	`artkeys="<Class1>=<Value1>|<Value2>,<Class2>=<Value3>`
	`<!--`
	`/* Font Definitions */`
	`@font-face`
	`{font-family:Verdana;`
	`panose-1:2 11 6 4 3 5 4 4 2 4;`
	`mso-font-charset:0;`
	`mso-generic-font-family:swiss;`
	`mso-font-pitch:variable;`
	`mso-font-signature:536871559 0 0 0 415 0;}`
	`/* Style Definitions */`
	`p.MsoNormal, li.MsoNormal, div.MsoNormal`
	`{mso-style-parent:"";`
	`margin:0in;`
	`margin-bottom:.0001pt;`
	`mso-pagination:widow-orphan;`
	`font-size:10.0pt;`
	`mso-bidi-font-size:12.0pt;`
	`font-family:Verdana;`
	`mso-fareast-font-family:"Times New Roman";`
	`mso-bidi-font-family:"Times New Roman";}`
	`@page Section1`
	`{size:595.45pt 841.7pt;`
	`margin:.7in .7in 2.4in .7in;`
	`mso-header-margin:.5in;`
	`mso-footer-margin:.5in;`
	`mso-paper-source:0;}`
	`div.Section1`
	`{page:Section1;}`
	`-->`

Cachetime	The option cachetime is used to determine how long a Agent should be cached. By default, cachetime is disabled. To use the cachetime option, you must run the tag uncached (Add !). The option is used like this: Cachetime=X where X is the number of minutes the list should be cached. `<pbs:!agent category="AP%" cachetime="10">`

Category

Determines the category from which stories should be retrieved. This option has no default but it **MUST** be set.
Option

```
Current= The category currently being
viewed.
Current%
= The category currently being viewed
with subcategories.
All
= All categories
%
= All categories
Categoryname
= A specific category
```

The category names available depends on your press agency feed. The following are options for various press agencies.

	• **AP** AP = The Associated Press (North America) http://www.ap.org. AP customers can use the following categories: AP% = All AP categories. APA = Nation APB = Nation/Special APC = Feature APD = Feature/Food APE = Feature/Entertainment APF = Business APG = Metro/Briefs API = Foreign APJ = Metro/Lottery APK = Op-Ed APL = Feature/Lifestyle APN = Metro APO = Weather APP = Nation/Politics APQ = Sports/Single APR = Sports/Racing APS = Sports APT = Feature/Travel APU = Metro APV = Advisory APW = Nation/Washington APX = Sports/Agate • **CP** CP = Canadian Press (Canada) http://www.cp.org. CP customers can use the following categories: CPB = Business CPS = Sports CPN = National CPE = Entertainment CPP = Panamerican Games CPW = World • **NTB** = Norsk Telegrambyr (Norway) http://www.ntb.no. NTB customers can use the following categories: NTBI = NTB Innenriks NTBU = NTB Utenriks NTBS = NTB Sport NTBO = NTB ¯konomi • **RZ** = Ritzaus Bureau (Denmark) http://www.ritzau.dk. RZ customers can use the following categories: RZI = Ritzau Indland RZU = Ritzau Udland RZK = Ritzau Kriminal RZS = Ritzau Sport RZE = Ritzau Erhverv RZP = Ritzau Politik • **TT** = Tidingarnas Telegrambyr AB (Sweden) http://www.tt.se/start/. TT customers can use the following categories: TTI = TT Inrikes TTU = TT Utrikes TTE = TT Ekonomi TTS = TT Sport TTP = TT Resultater ||
Count	Controls the number of stories that will be shown. By default, all stories modified today are shown. Enter any number between 1 and 999 in `count=x` to change the number of stories to be shown.

Keyword	If the news feed contains keywords (NTB and AP) you can extract stories based on keywords.To use keywords enter `Keyword=X` wher e X is the keyword. You can also do partial matching using the % sign like this: `Keyword=ol-%`. You can also do a partial match like this: `K eyword=%war%`. You cannot use multiple keywords.You can use
	
	`Category=AP[FP]`
	
	to display stories from both APF and APP. You can use multiple % signs so the string does not have to start with the keyword. You cannot use multiple keywords.The AP feed contains keywords for the state. To list all stories with the keyword *Alabama* you can use this tag:
	
	`<pbs:!agent category=AP% keyword=AL% cachetime=10>`

Leadin	Determines how many lead-in characters will be shown. By default, the entire lead-in is shown. Enter any number of characters from 0 to 9999 to show that number of lead-in characters. Setting this to 0 will turn off the lead-in option.
Location	Displays stories with a specific dateline. By default, this option is disabled. Entering `Location="X"` will show all stories with the specified dateline.The tag appears like this: `<pbs:!agent category=% location="New Delhi"cachetime=10>` Only stories listed in the specified dateline will be shown. You can further limit the output using the count option.
Maxpicturewidth	Determines the maximum width of the image.
Objectclass	Determines which object file should be used. By default, the general objects will be used. Object class can be set from 1 to 999 by entering `objectclass=x` where x is the number.
Priority	Shows stories matching a specific priority. By default, this option is disabled. If you enter `Priority=1` it will show all stories that have priority 1. You can specify only one priority for each tag. The priority codes differ from vendor to vendor, so some experimentation might be necessary.The tag then appears like this: `<pbs:!agent category=AP% priority=1 cachetime=10>`
Priosort	Sorts the story list by priority. By default, this option is disabled and stories are sorted by timestamp by default. Enter {{Priosort=1 }}to sort stories by priority. The result is that stories are sorted by priority code.

Start	Controls how many stories the tag should skip before it displays stories. This option can be used when you want to place an ad in the listing of stories or use multiple Newslist tags on the same page to control layout. By default, the system starts at the first story. You can set start story between 1 and 999 by entering the desired number in `start=x`

Theme	If the news feed contains a theme (AP) you can extract stories based on theme. By default this option is not used and all themes are listed.To use this option enter
	
	`Theme="X"`
	
	You can also do a partial match like this:
	
	`Theme="202%"`
	
	You cannot use multiple themes. A list of AP theme codes is available attached to this knowledge object.Theme is used like this in the tag:
	
	`<pbs:!agent category=AP% theme="1000" cachetime=10>`
	
	The result is that only stories with a matching theme will be output.

Topstories	Displays a list of stories based on the newest priority document from your press agency. By default, this option is disabled. Entering `Topstories=X` will show all stories for the Top story category. A list of AP theme codes is available at www.support.us.saxotech.com in the FAQ section.The tag appears like this: `<pbs:!agent category=% topstories=9000 cachetime=10>` Only stories listed in the newest priority document will be shown. You can further limit the output using the count option.

Agent.pbo

<%Items%>	This field outputs the items.

AgentItem.pbo

Used to list stories.

<%Counter%>	The item counter. The counter starts at 1 for the first item.
<%Date%>	The publication date preformatted.

<%DateHeader%>	The date header. The tag supports formatting, and can be used like this: `<%DateHeader$d="mmmm d."%>` It is only output when the date changes.
<%ImgLink%>	The image link. It must be used in an `<img src="<%ImgLink%>">` tag.
<%IsoDateTime%>	The date and time in ISO format. It can be formatted using object script.
<%Leadin%>	The lead-in.
<%Link%>	The link to the entire story.Example use: `<a href="<%Link%>">Linktext</a>`
<%Time%>	The publication time preformatted.
<%Title%>	The title for the story.

pbs agentcategory

<pbs:agentcategory>

This tag can be used on all **_art.pbs** templates on which you display agency stories.

pbs contentlist

<pbs:contentlist>

The **Contentlist** tag is used to build a list of mixed types (stories, galleries, and CCE objects) and can be used on all templates.

The content list always uses object files, namely the ContentList.pbo, and the ContentListItem.pbo files

General options

Execute	Holds whether to actually execute the tag (=1) or not (=0). By default the system will execute the tag (initial value is 1).
ObjectClass	Controls which object file will be used. Enter `Objectclass=1` to make the system look for ContentList1.pbo and ContentListItem1.pbo. The rules defining which object files will be used are described in the section About Object Files. You should also read the section on the folders in which the system will look for object files, which can be found in Directory structure.
Cachetime	This option controls how often the cache file for this tag should be updated. It is set in minutes and defaults is the general cachetime setting. The value of 0 means the tag is to run uncached. From Version 7.2.0sr38, 7.3.0sr3, this will default to uncached, instead of the cachetime setting.

Searchable	Holds a comma-separated list of tag parameters that the tag allows to be overridden through using matching URL parameters. E.g,
	
	`<pbs:contentlist searchable="text,profile,pagenumber" days="10" types="cce" classes="103">`
	
	will allow the users to free-text search and/or select a specific profile to search in using the URL's of
	
	`http://www.example.com/section/example?text=superbowl&profile=1134&pagenumber=2`
	
	All tag parameters except searchable itself are valid values for this parameter.

FacetFields	A comma separated list of fields where facets should be enabled. Possible values are: Keywords, TaxonomyLeaf, TaxonomyLeafId, Taxonomy, TaxonomyId, Rating, Location, Byline, Date, GeoName, GeoLocation, Channels, NumberImages, Category, Heading, Type, Types, SubTypes, DetailTypes, Summary, Module, Class, EventDate, TypeName, Year, Month, Day and CreatedTime.
FacetDates	A comma separated list of fields where facet dates should be enabled. Possible values are: Date, CreatedTime.
FacetDateStart	A start date in local time specified in UTC time format. This is used as fall back for all date facets. This is an example of a start date: 2011-08-20T00:00:00Z
FacetDateStart_{FieldName}	A specific start date used for one date facet.
FacetDateEnd	An end date in local time specified in UTC time format. This is used as fall back for all date facets. This is an example of an end date: 2011-08-20T00:00:00Z
FacetDateEnd_{FieldName}	A specific end date used for one date facet.
FacetDateGap	Gap or interval between start and and end for date facets. A number followed by HOUR, DAY, MONTH or YEAR. eg. %2B1DAY(+1DAY)
FacetDateGap_{FieldName}	A specific gap/interval used for one date facet.
FacetLimit	Specify the number of items returned in a facet in a comma separated list. `<pbs:contentlist FacetLimit="byline=7,category=2" facetfields="Category,Byline">`

Images	The parameter specifies a ordered list of image types which the system is restricted to output for each of the content items and in the given order. Not providing a value will make the system output all image types. The following example is the default value, so it includes all image types and in their default order: `<pbs:contentlist images="teaser,main,mainfactbox,paragraph,extra,gallery,logo,cceimage" />`

ListFields	Specifies a list of custom/extra fields whose values will be available in the item template. For example:
	
	`<pbs:contentlist days="2" listfields="color,saxonote">`
	
	will make two extra variables available in ContentListItem.pbo, namely:
	
	`<%ListField_Color%>, <%ListField_SaxoNote%>`
	
	The variables will have content from an extra field or custom field on all content types (articles, galleries, CCE) if they have an extra field with the name *Color* and/or *SaxoNote*.

PopulateEventVariables	Used when the cce events and recurrence variables are need in the output.

Filter options

AccessControlsExcl **From version 7.3.0sr1**	Access control exclude filter that filter the output to exclude items with at least one of the given access control items. Providing the filter value * will mean it will try to match all access control, outputting all items without any access controls.
AccessControlsIncl **From version 7.3.0sr1**	Access control include filter that filter the output to only include items with at least one of the given access control items. Providing the filter value * will mean it will try to match all access control, outputting all items with access controls.

Categories	Used when you want to filter content items on categories, but the filter applies only to stories and galleries. It is possible to use the pattern matching. For example,
	
	`categories="NEWS,SPORTS"`
	
	selects items from both the *News* and *Sports* categories as in this example:
	
	`categories="news%,sports0[1-3],arts[^4],opinion"`
	
	The options are:
	
	`categories="news%,sports%"`
	
	Selects items from all News and Sports categories.
	
	`categories="news0[5-6],sports0[1-2]"`
	
	Selects items from the categories news05, news06, sports01 and sports02. Please note that writing `Categories="CATEGORYID[05-06|05-06]"` is not supported.
	
	`categories="news0[^5],sport0[^1]"`
	
	Selects items from all categories except for news05 and sports01.

Count	How many content items to return. Default is 50, and maximum is 50. If PageSize is defined, PageSize is used instead of Count.

CreatedTime	In what date interval to search for content items. It compares with the created time. Predefined intervals are available: **LastYear**, **LastMonth**, **LastWeek**, **ThisYear**, **ThisMonth**, **ThisWeek** - but it is also possible to set it to a custom date range using the XSD Date Time for mat, for example:
	
	`<pbs:contentlist createdtime="lastweek">`
	`<pbs:contentlist createdtime="thismonth">`
	`<pbs:contentlist createdtime="2010-01-01T00:00:00Z,2010-12-31T23:59:59Z">`

CCETypes	Used when you want to filter CCE items based on type ids. Other content types are unaffected by this filter. `<pbs:contentlist types="cce" ccetypes="Type1,Type2"`
Classes	Used when you want to filter CCE items based on class ids. Other content types are unaffected by this filter. `<pbs:contentlist types="cce" classes="2,4,8">`

CreatedBy	List items filtered by the user specified.
	Frontend: CreatedBy="userid-xxxxx"
	Backend: Created="UserID"
	
	`<pbs:contentlist CreatedBy="Chris">`
	`<pbs:contentlist CreatedBy="userid-123456">`
	
	The tag will select only items submitted by this users.
	
	`<pbs:contentlist CreatedBy="#CurrentUser#">`
	
	The tag will retrieve current logged in user id from the session object and list items.

CreatorTypes	What types of owners allowed for extracted items. For now the following creator types are supported: user, mojo, edit. By default all are extracted. **user**: front-end users. **mojo**: front-end users that are mapped to an edit user. **edit**: edit users. `<pbs:contentlist creatortypes="mojo,edit">`
Date	Overrides **days** parameter and replaces it with a start and end time of when to search for stories so it can be used for listing evergreen items. Date is a comma-separated interval of dates supplied in XSD Date format - for example: `<pbs:contentlist date="2010-01-01Z,2010-12-31Z">`

Days	How many days back in time to look for content. Default is 7 days. This filter can also be disabled by giving the value of -1 (from version 6.7.0). Be aware when using the value -1, this will show articles using 99999999 as date. ***Warning: Giving a high value for this option or disabling it can drastically hamper performance***. ***We strongly suggest using pagination when using a high number of days.***
{FieldName}Search	This parameter will enable free text search in pre-defined or custom fields. Possible pre-defined field names are: Keywords, TaxonomyLeaf, TaxonomyLeafId, Taxonomy, TaxonomyId, Rating, Location, Byline, Date, GeoName, GeoLocation, Channels, NumberImages, Category, Heading, Type, Types, SubTypes, DetailTypes, Summary, Module, Class, EventDate, TypeName, Year, Month, Day and CreatedTime. When searching in custom fields the name used in the specific content type must be used. If a field exists in CCE with the name Color it can be searched by adding a parameter like this: `<pbs:contentlist ColorSearch="red">`
InternalModifiedTime	In what date interval to search for content items. It compares with the internal modified time by using the XSD Date Time format, for example: `<pbs:contentlist internalmodifiedtime="2010-01-01T00:00:00Z,2010-12-31T23:59:59Z">`
Keywords	Filters on stories' keywords, other content types are not affected by this filter. A comma separated list of keywords to filter on is expected and when looking for stories the filter will only let stories through that has any of the keywords given. Example: `<pbs:contentlist keywords="sports,olympics">`
Keys	You can filter on GUID by giving them as a comma list to this parameter. Example: `<pbs:contentlist keys="E7D3E8C8-511F-4352-9421-21B43BB946AF,02D7BD38-BEF7-42DE-A9D0-B70666F53B31">`
LinkDirections	You can output items being "children", "parents" or a comma separated list of them to specify the exact characteristics of the link filter "linkguid". Default is "children, parents". `<pbs:contentlist linkguid="<%ItemGUID%>" linkdirections="children" />`

LinkGUID	A filter on links to the contentitem with the given GUID. See parameter **LinkDirections** for providing exact characteristics of the link (parents and/or children) in relation to the contentitem with the given GUID. `<pbs:contentlist linkguid="<%ItemGUID%>" linkdirections="children" />`
ModifiedTime	In what date interval to search for content items. It compares with the modified time by usingthe XSD Date Time format. The difference between **modifiedtime** and **internalmodifiedtime** is that if the feature retain date for stories are used, the **modifiedtime** would be retained whereas the **internalmodifiedtime** still would reflect that the story was updated. Other content types do not have this feature. Examples of use: `<pbs:contentlist modifiedtime="2010-01-01T00:00:00Z,2010-12-31T23:59:59Z">`
Modules	Used when you want to filter CCE items based on module ids. Other content types are unaffected by this filter. `<pbs:contentlist types="cce" modules="2,4,8">`
PageNumber	What page of the search result to return. Default 1.
PageSize	How many content items to return per page. Maximum PageSize is 50 and default page size is 50. If you define the PageSize option, the Count option is ignored.

Profiles	You can output items matching a specific profile or profiles. Example:
	
	`<pbs:contentlist profiles="1142">`
	
	`<pbs:contentlist searchable="profile">`
	
	Takes the profile from the URL. Searchable=profile will override profile=x

QueryID	Given a custom query ID, then the requirements of the query is added to the filtering parameters. E.g., supplying a query that requires the profile 1011 and then have the profile=1000 parameter on the tag, then both profiles are required for items to appear in the result.

Relevance	Relevance can only be used in combination wtih relevance on sorting. Relevance is a way of calculating the quality of the content. All results returned is given a default score. When adding relevance as a sort parameter to contentlist it will sort content by a score that is calculated based on the fields where the search terms are located. The score calculated is also based on the freshness of the story. When sorting on relevance it will also be possible to choose from 3 different relevance functions: heading, text and byline. Heading will give a higher score to stories where the search term is found in the heading, subheader and netheader. Text is a mixture of title, summary, keywords, byline and paragraph text. Byline will give higher scores to stories where the search term is found in the byline. Relevance is not added to search terms used in field or group searches. `<pbs:contentlist sort="relevance desc" Relevance="text">`
Sites	Used to pull content from multiple sites codes. Used with the setting AllowedSiteCodes for papers with multiple site codes in the system. `<pbs:contentlist Sites="AA,BB">`
SynchID	This parameter will make the tag synchronize with other tags with the same SynchID value and make sure they do not display the same items at the same time - if the tags support the feature. For now that is: newslist, contentstats and other contentlist tags. `<pbs:contentlist synchid="1">`
TaxonomyWords	You can output items matching any of the given taxonomy words. Example: `<pbs:contentlist taxonomywords="12,14">`

Text	Free text search for the parameter value. By default it is not free text searching. Example:
	
	`<p>Stories on Obama</p>`
	`<pbs:contentlist types="stories" text="obama">`

Theme	Filters on stories' theme. Other content types are unaffected by this filter. Example: `<p>Olympics 2012</p> <pbs:contentlist types="stories" theme="olympics2012">`

Types	What types of content to extract. For now the following content types are supported: stories (stories), Photo galleries (galleries), and CCE objects (CCE). By default all are extracted. `<pbs:contentlist types="stories,galleries,cce">`

Sorting options

Sort	How to sort the list. The following values for this option are available:
	
	createdtime asc: Sort by oldest created items.
	createdtime desc: Sort by newest created items.
	modifiedtime asc: Sort by oldest modified items.
	modifiedtime desc: Sort by newest modified items.
	numcomments asc: Sort by least commented items.
	numcomments desc: Sort by most commented items.
	latestcomments asc: Sort by latest commented items. From versions 7.2.0sr40 and 7.3.0sr5
	latestcomments desc: Sort by latest commented items. From versions 7.2.0sr40 and 7.3.0sr5
	heading asc: Sort on title.
	heading desc: Sort on title.
	contenttype asc: Sort on type of content.
	contenttype desc: Sort on type of content.
	date asc: Sort on date.
	date desc: Sort on date.
	relevance asc: Sort on relevance From versions 7.1.0sr23 and 7.2.0sr15
	relevance desc: Sort on relevance From versions 7.1.0sr23 and 7.2.0sr15
	plainpriority asc: Sort by highest priority From versions 7.2.0sr23
	plainpriority desc: Sort on lowest priority From versions 7.2.0sr23
	profilepriority asc: Sort by highest priority on a single filtered profile From versions 7.2.0sr23
	profilepriority desc: Sort on lowest priority on a single filtered profile From versions 7.2.0sr23
	
	`<pbs:contentlist sort="createdtime desc">`
	`<pbs:contentlist sort="modifiedtime asc">`
	`<pbs:contentlist profiles="1000" sort="profilepriority desc">`
	
	The sort options also supports second sorting, example:
	
	`<pbs:contentlist sort="numcomments desc,date desc">`

Output statistics options

Channels	A filter on Channel IDs. Statistics gathered will be filtered so that only views from one of the given channels are included in the count. Example: `<pbs:contentlist interval="3days" channels="MOBILE,TABLET">`

Interval	Determines what interval to use when retrieving view statistics. The parameter is required for outputting view statistics. The following intervals are available: **15mins**, **30mins**, **60mins**, **120mins**, **5hours**, **12hours**, **24hours**, **48hours**, **3days**, **7days**, **30days**, **6months**, **12months**, **ThisWeek**, **LastWeek**, **ThisMonth**, **LastMonth**, **ThisYear**, **LastYear**, **AllTime**. Examples:
	`<pbs:contentlist interval="30mins">`
	`<pbs:contentlist interval="24hours">`
	`<pbs:contentlist interval="lastweek">`

Referrers	A filter on Referrer IDs. Statistics gathered will be filtered so that only views from one of the given referrers are included in the count. Example: `<pbs:contentlist interval="3days" referrers="GOOGLE,BING,YAHOO">`

ContentList.pbo

The base template for the **<pbs:ContentList>** tag.

The following variables are available:

List content

<%Items%>	This tag outputs all the items. The output is generated through the ContentListItem.pbo object file.

Counts

<%ItemCount%>	The number of items in the list.
<%TotalCount%> **From version 6.8.0**	The total number of items available in the list.
<%CCECount%>	The number of CCE objects in the list.
<%GalleryCount%>	The number of photo galleries in the list.
<%StoryCount%>	The number of stories in the list.
<%Facets%> **From version 6.8.0sr2**	Returns all facets.
<%Filters%> **From version 6.8.0sr2**	Returns all facet filters.

Example of pagination template

```
<pbs:!contentlist types="stories" text="<%q%>" days="-1" PageSize="<%SearchPageSize%>"
TaxonomyWords="<%TaxonomywordsSite%>" Searchable="pagenumber" PageNumber="1"
ObjectClass="3">
```

```
<%?SearchPageSize=10%>
<%CurrentRequestUnmodified$r("(^.*)(.pagenumber.\d+)(.*$)=$1$3")V(CurrentRequestMinusP
ageNumber)%>
<%CurrentRequestUnmodified$r("^.*pagenumber.(\d+).*$=$1")$r("^[^\d].*$=1")V(PageNr)%>
<%PageNr#*#SearchPageSize#$V(MaxCount)%>
<%SearchPageSize#-1$V(FromCount)%>

<!-- Start pagination -->
<div class="pagination">
  <div class="pagination-results">
  [%{MaxCount<#TotalCount#}
    Show <%MaxCount#-#FromCount#%>-<%MaxCount%> of <%TotalCount%>
  %|%
    Show <%MaxCount#-#FromCount#%>-<%TotalCount%> of <%TotalCount%>
  %]
  </div>
  <span class="previous">
    [%{PageNr>1}<a
href="<%CurrentRequestMinusPageNumber%>&PageNumber=<%PageNr#-1%>">&laquo;
previous</a>%]
  </span>
  <span class="next">
    [%{MaxCount<#TotalCount#}<a
href="<%CurrentRequestMinusPageNumber%>&PageNumber=<%PageNr#+1%>">next &raquo;</a>%]
  </span>
</div>
<!-- End pagination -->
<%Items%>
```

Or with numbers (< previous | 1 | 2 | 3 | next >)

```
<%CurrentRequestUnmodified$r("^.*pagenumber.(\d+).*$=$1")$r("^[^\d].*$=1")V(Custom_Pag
eNr)%>
<%Custom_PageNr#*10$V(Custom_MaxCount)%>
<div class="sumsort">
  <p>
    [%{Custom_MaxCount<#TotalCount#}
      Showing <%Custom_MaxCount#-9%>-<%Custom_MaxCount%> articles of <%TotalCount%>
    %|%
      Showing <%Custom_MaxCount#-9%>-<%TotalCount%> articles of <%TotalCount%>
    %]
  </p>
</div>
<div class="results" id="ajaxResults">
  <%Items%>
</div>
<div class="pagination" id="pagination"></div>
<script>
function is_float (mixed_var) {
```

```
  return +mixed_var === mixed_var && (!isFinite(mixed_var) || !!(mixed_var % 1));
}
(function()
{
  LTotalPages = (<%TotalCount%>/10);
  LHtml = '<h2>Pages:</h2><ul>';
  LCurrentPage = <%Custom_PageNr%>;
  LUrl = '/section/industryinsights/<%profile_id%>&PageNumber=';
  LCnt= 0;
  LNewStart = 0;
  if(is_float(LTotalPages))
      LTotalPages = parseInt(LTotalPages)+1;
  if(LCurrentPage != 1)
    LHtml += '<li><a href="'+LUrl+(LCurrentPage-1)+'">Prev</a></li>';


  // Before the current page
  if(LCurrentPage-6 <= 1){
    while (LCnt+1<LCurrentPage){
      LCnt++;
      LHtml += '<li><a href="'+LUrl+LCnt+'">'+LCnt+' </a><li>';
    }
    LCnt++;
    LHtml += '<li>' + LCnt+'</li>';
  }
  else{
    LHtml += '<li><a href="'+LUrl+1+'">'+1+'... </a></li>'; //first page
    LNewStart = LCurrentPage-6;
    while (LNewStart<LCurrentPage-1){
      LNewStart++;
      LHtml += '<li><a href="'+LUrl+LNewStart+'">'+LNewStart+' </a></li>';
    }
    LNewStart++;
    LHtml += '<li>' + LNewStart+'</li>'; // current page
  }


  // After the current page
  if(LCurrentPage+5 > LTotalPages){
    LCnt = LCurrentPage;
    while (LCnt<LTotalPages){
      LCnt++;
      LHtml += '<li><a href="'+LUrl+LCnt+'">'+LCnt+' </a></li>';
    }
  }
  else{
    LNewStart = LCurrentPage;
    while (LNewStart<LCurrentPage+5){
      LNewStart++;
      LHtml += '<li><a href="'+LUrl+LNewStart+'">'+LNewStart+' </a></li>';
    }
    LHtml += '<li><a href="'+LUrl+LTotalPages+'">...'+LTotalPages+' </a></li>'; //
last page
  }


  if(LCurrentPage < LTotalPages)
    LHtml += '<li><a href="'+LUrl+(LCurrentPage+1)+'">Next</a></li>';
  LHtml += '</ul>';
```

```
$('#pagination').html(LHtml);
```

```
}
)();
</script>
```

ContentListByline.pbo

This object file outputs the byline of each item content list.

The variables available are exactly the same as for the FreestyleStoryItemByline.pbo object file.

ContentListFacetItem.pbo

Template is available from version 6.8.0sr2
Used for facets in content list.

<%Counter%>	Outputs the current item's number in the list.
<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	The number of hits related to the current modifier.
<%ModifierValue%>	Returns a pre-made search string.
<%ModifierDateFrom%>	The from date specified for a date facet.
<%ModifierDateTo%>	The to date specified for a date facet.
<%ModifierDateGap%>	The gap/interval specified for a date facet.
<%FacetName%>	The name of the facet.
<%FacetType%>	Type of facet. Possible values are date and field.

ContentListFacetList.pbo

Template is available from version 6.8.0sr2
Used for preparing search in stories.

<%FacetName%>	The name of the current facet.
<%FacetType%>	Specifies the type of facet. Possible values are date and field.
<%Items%>	Holds all elements (modifiers) for one facet. All modifiers use the ContentListFacetItem.pbo file.
<%Count%>	Number of elements in the facet.

ContentListFilterItem.pbo

Template is available from version 6.8.0sr2
Used for generating each individual selected filter.

<%FilterName%>	The name of the filter item.
<%FilterValue%>	Value used in selected filter.
<%RemoveFilterUrl%>	Remove the selected filter from the search result.

ContentListFilterList.pbo

Template is available from version 6.8.0sr2
Holds all the selected Facet filters.

<%Items%>	All the filter items.

ContentListItem.pbo

Object file used by **<pbs:contentlist>** tag to output the individual content items in the list. The final product of this file is output as a variable in the ContentList.pbo object file.

The following variables are available:

List navigators

<%Counter%>	An incremental item counter, going from 1 to **TotalCount**
<%IsLastItem%>	Has the value 1 if the item is the last item, i.e., **Counter** has same value as **TotalCount**.
<%TotalCount%>	Holds the total count of items in the list being displayed.

Custom / extra field descriptors

ListField_{FieldName}	Specifies a list of custom / extra field whose values should be available in the item template. The example tag
	`<pbs:contentlist days="2" listfields="color,saxonote">`
	will make two extra variables available in ContentListItem.pbo namely
	`<%ListField_Color%>, <%ListField_SAXoNote%>`
	The variables will have content from an extra field or custom field on all content types if they have an extra field with the name "Color" and/or "SAXoNote".

<%TaxonomyList%>	A list of taxonomy words on the item. This is rendered by the object file of ContentListItemTaxonomy.pbo. This variable is also available via the pbs:contentstats tag - the object file from above is used.

Item ID descriptors

<%Site%>	The sitecode the content item belongs to.
<%Date%>	The date of the content item.
<%Category%>	The category of the story or gallery.
<%CategoryName%> **From version 7.2.0sr30**	The story or gallery's category display name.
<%ID%>	The lopenr of stories and galleries or the object ID of the CCE object.
<%GUID%>	The GUID of the content item.
<%Module%>	The module ID of the CCE object.
<%ModuleName%>	The module display name of the CCE object.
<%Class%>	The class ID of the CCE object.
<%ClassName%>	The class display name of the CCE object.
<%CCEType%>	The main type ID of the CCE object.

<%CCETypeName%>	The main type display name of the CCE object.

Item descriptors

<%ContentLink%>	Holds the direct link for viewing the content item.
<%Title%>	The title of the content item.
<%NetTitle%> **From versions 7.0.0sr17, 7.1.0sr6**	The net title of the content item. Only stories have a net title.
<%TeaserTitle%> **From versions 7.0.0sr17, 7.1.0sr6**	The teaser title of the content item. Only stories have a teaser title.
<%Summary%>	The summary of the content item. Summary for stories. Text for CCE and Description for galleries.
<%Type%>	Contains the type of the item: "story", "gallery", or "cce".
<%DesignClass%> **From version 7.2.0sr23**	The design class of the story or gallery.
<%MainProfileID%> **From version 7.2.0sr30**	The ID of the item's main profile.

Story specific descriptors

<%Byline%>	Byline of the story.
<%SEOLabel%> **From version 7.0.0sr11**	SEO label of the story.
<%SmartByline%>	For stories with smartbyline activated it is possible to use the smartbyline variable to output a generated byline. The object file ContentListByline.pbo or ContentStatsByline.pbo is used.
<%FirstParagraphBody%>	First paragraph of the story.

Item images

<%ImageLink%>	Holds a link for viewing the image of the content item.
<%AltText%>	Holds the alt text for the item images that supports it.
<%Images%> From versions 7.1.0sr13 and 7.2.0sr2	The item's images, each formatted through the object file of ContentListItemImageItem.pbo Their order is always the same given the content item type: Story: **teaser**, **main**, **mainfactbox**, **paragraph** (by their individual order), **extra** (by their individual order) Gallery: **gallery** (by their individual order) CCE: **logo**, **cceimage** (by their individual order) Use the images tag parameter for pbs:contentlist to restrict what images to output.
<%ImageCount%> From versions 7.1.0sr13 and 7.2.0sr2	The total number of images output for the content item.
<%ImageHeight%> From version 7.2.0sr24	The height of the image in pixels.
<%ImageWidth%> From version 7.2.0sr24	The width of the image in pixels.
<%ImageOrientation%> From version 7.2.0sr24	The orientation of the image. Value is "L" or "P" - for landscape and portrait, respectively.
<%TeaserImageLink%> **From version 7.3.0sr1**	Holds a link for viewing the teaser image of the content item. Teasers are currently only supported by stories.

<%TeaserAltText%> **From version 7.3.0sr1**	Holds the alt text for the teaser image.
<%TeaserImageHeight%> From version 7.3.0sr1	The height of the teaser image in pixels.
<%TeaserImageWidth%> From version 7.3.0sr1	The width of the teaser image in pixels.
<%TeaserImageOrientation%> From version 7.3.0sr1	The orientation of the teaser image. Value is "L" or "P" - for landscape and portrait, respectively.

Created, Last modified, and Published descriptors

<%ISOCreatedDateTime%>	Created time of the item. Output format is yyyymmddhhnnss and can be reformatted using the $d modifier.
<%CreatedByUserType%>	Holds the type of the user output. "user", "mojo" or "edit"
<%CreatedByUserID%>	Contains the UserID of the creator. Value format depends on the **<%CreatedByUserType%>**: If the user type is a "user" then the full UserID-xyz is output. If it is a "mojo" or "edit", then the mapped edit user's username is output.
<%CreatedByName%>	Holds the name of the user if the user type is "mojo" or "edit"
<%CreatedByFirstName%>	Holds the first name of the user if the user type is "user" or "mojo"
<%CreatedBySurName%>	Holds the sur name of the user if the user type is "user" or "mojo"
<%CreatedByNickName%>	Holds the nickname of the user if the user type is "user" or "mojo"

<%ISOLastModifiedDateTime%>	Last modified time of the item. Output format is yyyymmddhhnnss and can be reformatted using the $d modifier.
<%LastModifiedByUserType%>	Holds the type of the user output. "user", "mojo" or "edit"
<%LastModifiedByUserID%>	Contains the UserID of the user doing the last modification. Value format depends on the **<%LastModifiedByUserType%>**: If the user type is a "user" then the full UserID-xyz is output. If it is a "mojo" or "edit", then the mapped edit user's username is output.
<%LastModifiedByName%>	Holds the name of the user if the user type is "mojo" or "edit"
<%LastModifiedByFirstName%>	Holds the first name of the user if the user type is "user" or "mojo"
<%LastModifiedBySurName%>	Holds the sur name of the user if the user type is "user" or "mojo"
<%LastModifiedByNickName%>	Holds the nickname of the user if the user type is "user" or "mojo"

<%ISOPublishedDateTime%> **From version 7.1.0sr12**	Published time of the item. Output format is yyyymmddhhnnss and can be reformatted using the $d modifier.

Comments and Ranking descriptors

<%AvgRanking%>	Contains the average ranking of the content item.
<%NumComments%>	Contains the number of comments on the item.
<%NoComments%>	Contains the number of comments on the item. **(Deprecated)**
<%TotalNoOfVotes%>	Contains the total number of votes on the item.
<%UserCommentsStatus%>	Holds whether comments are allowed on the item or not. 1 = yes, 0 = no.

View statistics descriptors

<%Channel_(ChannelID)_Views%>	Holds the number of views through the channel in the selected interval. Example: `<%Channel_MOBILE_Views%>`
<%Referrer_(ReferrerID)_Views%>	Holds the number of views from the referrer in the selected interval. Example: `<%Channel_BING_Views%>`
<%Views%>	Holds the total number of views in the selected interval.

Access Control descriptors

<%AccessControlIdentifier%>	Indicates the flow ID that is associated with a story or gallery in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control. The variable <%CurrentFlowItemIdent%> should be used from version 7.1.0 instead, as this also work with CCE items.
<%CurrentFlowItemId%> From version 7.1.0	Represents the current access element in the flow of the current item as an ID number. This variable is used in conjunction with Access Control.
<%CurrentFlowItemIdent%> **From version 7.1.0**	Represents the current access element in the flow of the current item as the identifier. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%> **From version 7.1.0**	Outputs the standard text field of the current access element in the flow of the current item. This variable is used in conjunction with Access Control.
<%SingleSaleId%> **From version 7.1.0**	This field contains the id of the access control element attached to the item.
<%SingleSaleIdent%> From version 7.1.0	This field contains the identifier of the access control element attached to the item.
<%SingleSaleText%> From version 7.1.0	This field contains the text of the access control element attached to the item.
<%AccessControls%> From version 7.3.0sr1	Outputs the access control items using the object file of ContentListItemAccessControls.pbo

Displaying Event objects

From version 7.0.0sr16 and 7.1.0sr4

The variables in this section are only available if the contentlist tag has the PopulateEventVariables="1" parameter and outputs CCE objects.

<%EventDate%>	Contains the first coming occurrence of the object.
<%EventDateList%>	Contains a list of the first many object occurrence dates. The dates are formatted as yyyymmdd,yyyymmdd....
<%EventToDate%>	Contains the date to which the system is to calculate occurrences (yyyymmdd format) - it is used when **Stop on** is set in Edit UI.
<%NumberOfOcc%>	Contains how many occurrences to calculate from the **EventFromDate** - it is used when **Stop after** is set in Edit UI.
<%StopMethod%>	Contains how the system is to stop calculating occurrences. 0 is on a set date, 1 is after a certain number of occurrences.

<%EventFromTime%>	Contains from what time of day the object occurs. The format of this variable has changed to 00000000HHMM which means that users can now use $d modifier to format the value this variable (<%EventFromTime$d("hh:mm AM/PM")%>).
<%EventToTime%>	Contains to what time of day the object occurs. The format of this variable has changed to 00000000HHMM which means that users can now use $d modifier to format the value this variable (<%EventFromTime$d("hh:mm AM/PM")%>).
<%Pattern%>	Contains the overall method used to calculate occurrences.
<%Subpattern%>	Contains the specific method used to calculate occurrences.
<%Jump1%>	Contains the first specific value in the pattern.
<%Jump2%>	Contains the second specific value in the pattern.
<%ActiveDays%>	Contains what days in the week are selected in the pattern.
<%Day1%>	If Sunday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day2%>	If Monday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day3%>	If Tuesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day4%>	If Wednesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day5%>	If Thurday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day6%>	If Friday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day7%>	If Saturday is in the **ActiveDays**, then it has value 1, otherwise it has no value.

Displaying Asset objects **From version 7.2.0**

The variables in this section are only available if the object is a CCE item and it belongs to an Asset module.

<%External%>	Shows if the asset is external. Values (0,1)
<%AssetTypeID%>	ID of the assettype connected to the CCE item
<%AssetTypeName%>	Name of the assettype connected to the CCE item
<%VirtualFolderID%>	ID of virtual folder used by the asset
<%VirtualFolderName%>	Name of the virtual folder.
<%VirtualFolderFilePath%>	FilePath for the virtual folder
<%VirtualFolderStatus%>	Status of the virtual folder. (protected, public)
<%FileSize%>	Size of the asset file in bytes
<%FileName%>	Name of the asset file
<%AssetURL%>	URL to the binary file

ContentListItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of ContentListItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

ContentListItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

ContentListItemImageItem.pbo

⚠ **The template is available from versions 7.1.0sr13 and 7.2.0sr2**

Used for rendering images for the contentlist and contentstats tags for the <%Images%> variable in ContentListItem.pbo.

Image Data

<%ImageGUID%>	The GUID of the image.
<%ImageRef%>	The ref of the image.
<%ImageType%>	The type of the image. One of the following values will be returned based on what kind of image it is: main, teaser, mainfactbox, paragraph, extra, gallery, logo, cceimage
<%Title%>	The title of the image.
<%Body%>	The caption of the image.
<Byline>	The byline of the image.
<%AltText%>	The alternative text of the image to be used in case the browser does not display the image for some reason.
<%Align%>	The alignment of the image. This will be either 'left' or 'right' for paragraph images.
<%ImageLink%>	The link to the image.
<%ImageHeight%> **From version 7.2.0sr24**	The height of the image in pixels.
<%ImageWidth%> **From version 7.2.0sr24**	The width of the image in pixels.
<%ImageOrientation%> **From version 7.2.0sr24**	The orientation of the image. Value is "L" or "P" - for landscape and portrait, respectively.

Counting

<%Counter%>	Contains the counter value (image number) for the image.
<%TotalCount%>	The total number of images being outputted.
<%IsLast%>	Has the value of 1 if the image is the last one being rendered.

Example

```
<table>
  <tr><td>Counter</td><td><%Counter%></td></tr>
  <tr><td>TotalCount</td><td><%TotalCount%></td></tr>
  <tr><td>IsLast</td><td><%IsLast%></td></tr>
  <tr><td>ImageGUID</td><td><%ImageGUID%></td></tr>
  <tr><td>ImageRef</td><td><%ImageRef%></td></tr>
  <tr><td>ImageType</td><td><%ImageType%></td></tr>
  <tr><td>Title</td><td><%Title%></td></tr>
  <tr><td>Body</td><td><%Body%></td></tr>
  <tr><td>Byline</td><td><%Byline%></td></tr>
  <tr><td>AltText</td><td><%AltText%></td></tr>
  <tr><td>Align</td><td><%Align%></td></tr>
  <tr><td>ImageLink</td><td><%ImageLink%></td></tr>
</table>
```

ContentListItemTaxonomy.pbo

<%Items%>	Outputs the taxonomy items using the object file of ContentListItemTaxonomyItem.pbo
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

ContentListItemTaxonomyItem.pbo

<%ID%>	Outputs the ID of the taxonomy word.
<%Description%>	Outputs the taxonomy word text.
<%Path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to separate items in the tree.
<%PathItems%>	Uses ContentListItemTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

ContentListItemTaxonomyItemPathItem.pbo

<%Parent%>	Outputs the description of the parent taxonomy word.

<%IsLast%>	Outputs 1 if the item is the last in the tree.
<%IsFirst%>	Outputs 1 if the item is the first in the tree.

pbs contentstats

<pbs:contentstats>

The **Contentstats** tag is used to build a list of mixed types (stories, galleries, and CCE objects) sorted on their view statistics and can be used on all templates.

The contentstats tag uses the same object files as the contentlist tag, namely the ContentList.pbo, and the ContentListItem.pbo files

Prerequisites
For ContentStats to work some initial setup needs to be done. DEV-Online6.7.0ReleaseNotes-110912-0153-270.pdf

General options

ObjectClass	Controls which object file will be used. Enter `Objectclass=1` to make the system look for ContentList1.pbo and ContentListItem1.pbo. The rules defining which object files will be used are described in the section About Object Files. You should also read the section on the folders in which the system will look for object files, which can be found in Directory structure.
Cachetime	This option controls how often the cache file for this tag should be updated. It is set in minutes and defaults is the general cachetime setting. The value of 0 means the tag is to run uncached. From Version 7.2.0sr38, 7.3.0sr3, this will default to uncached, instead of the cachetime setting.

Filter options

Categories	Used when you want to filter content items on categories, but the filter applies only to stories and galleries and not CCE objects. It is possible to use the pattern matching. For example,
	
	`categories="NEWS,SPORTS"`
	
	selects items from both the *News* and *Sports* categories as in this example:
	
	`categories="news%,sports0[1-3],arts[^4],opinion"`
	
	The options are:
	
	`categories="news%,sports%"`
	
	Selects items from all News and Sports categories.
	
	`categories="news0[5-6],sports0[1-2]"`
	
	Selects items from the categories news05, news06, sports01 and sports02. Please note that writing `Categories="CATEGORYID[05-06|05-06]"` is not supported.
	
	`categories="news0[^5],sport0[^1]"`
	
	Selects items from all categories except for news05 and sports01.

Channels	A filter on Channel IDs. Statistics gathered will be filtered so that only views from one of the given channels are included in the count. Example: `<pbs:contentstats interval="3days" channels="MOBILE,TABLET">`
Count	How many content items to return. Default is 10 items.

CreatedTime	In what date interval to search for items. It compares with the created time. Predefined intervals are available: **LastYear**, **LastMonth**, **Last Week**, **ThisYear**, **ThisMonth**, **ThisWeek** - but it is also possible to set it to a custom date range using this format: yyyymmdd,yyyymmdd. Examples:
	
	`<pbs:contentstats interval="3days" createdtime="lastweek">`
	`<pbs:contentstats interval="3days" createdtime="thismonth">`
	`<pbs:contentstats interval="3days" createdtime="20100101,20120101">`

Days	How many days back in time to look for content. Default is 7 days. This filter can also be disabled by giving the value of -1. ***Warning: Giving a high value for this option or disabling it will drastically hamper performance***.
Images **From versions 7.1.0sr13 and 7.2.0sr2**	The parameter specifies a ordered list of image types which the system is restricted to output for each of the content items and in the given order. Not providing a value will make the system output all image types. The following example is the default value, so it includes all image types and in their default order: `<pbs:contentstats interval="30mins" images="teaser,main,mainfactbox,paragraph,extra,gallery,logo,cceimage" />`

Interval	Determines what interval to use when looking up view statistics. It is a **required** parameter as the tag sorts by the statistics. The following intervals are available: **15mins**, **30mins**, **60mins**, **120mins**, **5hours**, **12hours**, **24hours**, **48hours**, **3days**, **7days**, **30days**, **6months**, **12months**, **ThisWeek**, **LastWeek**, **ThisMonth**, **LastMonth**, **ThisYear**, **LastYear**, **AllTime**. Examples:
	`<pbs:contentstats interval="30mins">`
	`<pbs:contentstats interval="24hours">`
	`<pbs:contentstats interval="lastweek">`

Profile	You can output items matching a specific profile. Example: `<pbs:contentstats interval="3days" profile="1142">`
Referrers	A filter on Referrer IDs. Statistics gathered will be filtered so that only views from one of the given referrers are included in the count. Example: `<pbs:contentstats interval="3days" referrers="GOOGLE,BING,YAHOO">`

Searchable	Holds a comma-separated list of tag parameters that the tag allows to be overridden through using matching URL parameters. E.g,
	
	`<pbs:contentstats interval="3days" searchable="profile" types="cce" classes="103">`
	
	will allow the URL to determine the profile to filter on
	
	`http://www.example.com/section/example?profile=1134`
	
	All tag parameters except searchable itself are valid values for this parameter.

SynchID **From versions 7.1.0sr13 and 7.2.0sr2**	This parameter will make the tag synchronize with other tags with the same SynchID value and make sure they do not display the same items at the same time - if the tags support the feature. For now that is: newslist, contentlist and other contentstats tags. `<pbs:contentstats interval="30mins" synchid="1">`
TaxonomyWords	You can output items matching any of the given taxonomy words. Example: `<pbs:contentstats interval="3days" taxonomywords="12,14">`
Types	What types of content to extract. For now the following content types are supported: stories (stories), photo galleries (galleries), and CCE objects (CCE). By default all are extracted. `<pbs:contentstats interval="3days" types="stories,galleries,cce">`

ContentStats.pbo

The base template for the pbs contentstats tag. If it is not present, the system will revert to use the ContentList.pbo object file if available.

The variables available are the same as the ones for ContentList.pbo

ContentStatsItem.pbo

The base template for the pbs contentstats tag. If it is not present, the system will revert to use the ContentListItem.pbo object file if available.

Object file used by pbs contentstats tag to output the individual content items in the list. The final product of this file is output as a variable in the ContentStats.pbo object file.

The variables available are the same as the ones for ContentListItem.pbo

pbs freestyle

From version 6.6.0sr20 the freestyle tag must be used instead of storylist tag.

DO NOT PUT STYLING ON THE BODY TAG. Otherwise editors will not be able to access the story through the edit interface.

<pbs:freestyle>

The **Freestyle** tag is used to build overviews of stories and can be used on all templates.

The freestyle storylist always uses object files, namely the FreestyleStoryList.pbo, FreestyleStoryItem.pbo, FreestyleStoryItemByline.pbo files

The template is a more specialized way than the **Newslist tag** to extract stories, and the method it employs is determined by its **type** parameter.

General options

ObjectClass	Controls which object file will be used. Enter `Objectclass=1` to make the system look for FreestyleStoryList1.pbo and FreestyleStoryItem1.pbo. The rules defining which object files will be used are described in the section About Object Files. You should also read the section on the folders in which the system will look for object files, which can be found in Directory structure.
Type	For now only the type "story" is supported. It used for Freestyle Publishing. **Story**: A list of stories that is controlled completely from the edit interface (Requires the Freestyle (Freestyle Publishing)module) `<pbs:freestyle type="story">`

Type descriptions

Story (Freestyle)

Freestyle storylists are lists that editors can create and manage manually from the edit interface. The GUI for editing the list of stories supports drag-and-drop from the story list to the edited list's positions and between positions. Also, custom titles and summaries can be given to the list items through a WYSIWYG editor.

Because of the drag-and-drop functionality, certain restrictions apply to the HTML in the list:

1. All story items must be surrounded by a tag.
2. The story list cannot be built using HTML tables.
3. The page showing the storylist tag must not contain elements with style position: Fixed.

The Story type additionally supports the following tag parameters:

KeywordLinksCount	This option tells the system how many stories with the same keywords should be listed. By default 5 stories are shown. To show two stories, enter `KeywordLinksCount=2`

KeywordLinksMonths	This option controls how many months back the system should look for stories with the same theme. By default the system looks back 3 months. To look back six months, enter: `KeywordLinksMonths=6`
Placement	The name of the freestyle list of stories the tag extracts. `<pbs:freestyle type="story" placement="frontpageleft" objectclass="2">`
Teaser	Show stories' teasers (=1) or not (=0).
ThemeLinksCount	This option tells the system how many stories with the same theme should be listed. By default, 5 stories are shown. To show two stories, enter: `Themelinkscount=2`
ThemeLinksMonths	This option controls how many months back the system should look for stories with the same theme. By default, the system looks back 3 months. To look back six months, enter: `ThemeLinksMonths=6`

ListFields
From version 7.2.0sr11
	Specifies a list of custom/extra fields whose values will be available in the item template. For example:
	`listfields="color,saxonote"`
	will make two extra variables available in FreeStyleListItem.pbo, namely:
	`<%ListField_Color%>, <%ListField_SaxoNote%>`
	The variables will have content from an extra field or custom field on all content types (articles, galleries, CCE) if they have an extra field with the name *Color* and/or *SaxoNote*.

3rd party integration to freestyle pages in Online

When a user is logged into Online as an edit user it is possible to get a freestyle storylist for edit in an external system using a template with custom scripts. In order to use a custom script template in the freestyle page, the call must be made with the following query parameters:

```
&handheldtemplate=Path+NameOfTemplate
```

The templates must be placed in the **Objects** folder or a subfolder to it on the frontend.

FreestyleStoryArticleConnectionItem.pbo

Used with the field `<%ArticleConnectionItems%>` in FreestyleStoryItemInternalLinks.pbo.

The file uses the same fields as FreestyleStoryItemConnectionItem.pbo except <%IS...%> fields are not available.

FreestyleStoryItem.pbo

> From 6.8.0sr6 Freestyle object item templates has changed. Old FreeStyle templates will therefor not work anymore. In FreestyleStoryItem two new variables have been added <%DivAttributes%> and <%DivDraggableClasses%>. Both variables must be included in a div tag on the item templates. This div tag must encapsulate all story related variables.

The following list of parameters is available for use in the object file. The output is used for the **<%Items%>** variable in the FreestyleStoryList.pbo object file.

<%Byline%>	The byline on the story. If using smart byline then output is generated through the FreestyleStoryItemByline.pbo object file.
<%Count%>	This field contains the total number of stories in the list.

<%Counter%>	This field starts with the value 1 and is increased by one for each story.
	`[%{counter=1}Story 1%]`
	`View ad 1`
	`[%{counter=2|counter=3}Story 2 and 3%]`

<%DivAttributes%>	A variable that must be included in a div tag. like this: `<div <%DivAttributes%>` `class="<%DivDraggableClasses%>">` `[%{ImageLink}` `<a href="<%Link%>"> <img src="<%Imagelink%>&MaxW=475&ExactW=475&ExactH=190" border="0"></a>  ` `%]` `<strong><a href="<%link%>"><h1><%title%></h1></a></strong>` `<%Summary%> ` `</div>`
<%DivDraggableClasses%>	A variable that must be included in the class attribute inside a div tag. Additional classes may also be included in the div tag
<%ImageLink%>	The link to the main story photo. You use the field like this: `[%{Imagelink}<img src="<%Imagelink%>&MaxW=120&MaxH=80" border=0 hspace=1 vspace=1 alt="" align="right">%]` The options for setting the maximum width and height of the image have also been added to the link. You can use either one or both.

<%IsoPubDateTime%>	The publication date and time in ISO format (YYYYMMDDHHMMSS). This field can be formatted using object script. It is written: `<%IsoPubDateTime%>`
<%IsEven%>	Whether the **Counter** is even.
<%IsOdd%>	Whether the **Counter** is odd.
<%Link%>	The URL to the story.
<%SectionCat%>	The ID for the category in which the story is shown. SectionCat is a flexible way to specify which template the page should use (without affecting the articles you get from the database).
<%ProfileID%>	The profile ID of the context in which the list is shown.
<%SkipForDraggedStories%> **From Version 7.2.0sr18**	This variable can be used to skip parts of the template, when dragging stories around in the edit interface. This is used to remove elements which is not part of the draggable stories in the template. This could be ads or headings between elements in the list.
<%StoryCategory%>	The category ID for the category in which the story is shown.
<%StoryDate%>	The date the story was last modified. This field uses the **Dateformat** option in the tag for formatting.
<%StoryGUID%>	The story's GUID in the database.
<%StoryID%>	The article ID, which can be used with category and date as here:`http://www.yoursite.com/apps/pbcs.dll/article?AID=/20060721/PCSPORT/607210374`
<%StoryRef%>	Contains the story reference, which differs from <%ImageRef%> when using teasers
<%StorySite%>	Contains the site code of the story.
<%Summary%>	This is the summary of the story. If the story does not have a summary, the first paragraph will be used.
<%Theme%>	Outputs the theme field for the story.
<%Title%>	The story title.

If **<%Link%>** is extracted then the following are available:

<%TeaserUrlTarget %>	Contains target on teaser url.

If **<%ImageLink%>** is extracted then the following are available:

<%AltText%>	The alternative text for an image if it is set.
<%ImageHeight%>	The height of the image.
<%ImageOrientation%>	The image orientation. The field has the value P for portrait and L for landscape. It is written: `<%ImageOrientation%>`
<%ImageWidth%>	The width of the image.
<%Caption%>	A predefined caption string based on **CaptionHead**, **CaptionBody**, and **CaptionByline**.
<%CaptionBody%>	The main text for the picture.

<%CaptionByline%>	The byline for the photo.
<%CaptionHead%>	The heading for the picture.
<%SeoFilename%>	The SEO friendly filename specified for the image if any is set.
<%UseIndependentImageVersion%> **From version 6.9.0sr13**	Has a value if the publicus.ini setting of [IMAGE] FreestyleUseStandardImageVersions=1 is NOT used. This is done so that migrating a site from using Freestyle's feature of providing its own imageversion for an image to using standard image crops. Once the site is fully migrated any object script conditionals using this variable can be removed again.
<%NumComments%> **From version 7.2.0sr11**	Contains the number of comments on the item.
<%UserCommentsStatus%> **From version 7.2.0sr13**	Outputs 1, 3 or is empty. Value 1 means that usercomments are activated on the story. Value 3 means that usercomments are locked on the story. If the variable is empty, this means that usercomments are not activated on the story.

ListFields
From version 7.2.0sr11
	Specifies a list of custom/extra fields whose values will be available in the item template. For example:
	
	`listfields="color,saxonote"`
	
	will make two extra variables available in FreeStyleListItem.pbo, namely:
	
	`<%ListField_Color%>, <%ListField_SaxoNote%>`
	
	The variables will have content from an extra field or custom field on all content types (articles, galleries, CCE) if they have an extra field with the name *Color* and/or *SaxoNote*.

If **<%ProfileID%>** is extracted then the following are available:

<%ProfileName%>	The name of the profile if profiles are used.
<%RelevantProfileID%>	The most relevant profile's ID.
<%RelevantProfileName%>	The name of the most relevant profile.

From 6.6.0 the object file holds the following variables.

<%AltText%>	The alternative text for an image if it is set.
<%Byline%>	The byline on the story. If using smart byline then output is generated through the FreestyleStoryItemByline.pbo object file.
<%Caption%>	A predefined caption string based on **CaptionHead**, **CaptionBody**, and **CaptionByline**.
<%CaptionBody%>	The main text for the picture.
<%CaptionByline%>	The byline for the photo.
<%CaptionHead%>	The heading for the picture.
<%CategoryDescription%>	The description field of the story's category.
<%CategoryInfo%>	The information field of the story's category.
<%CategoryLink%>	The link to the section front relating to the story's category.

<%CategoryName%>	The display name of the story's category.
<%Count%>	This field contains the total number of stories in the storylist.

<%Counter%>	This field starts with the value 1 and is increased by one for each story.
	
	`[%{counter=1}Story 1%]`
	`View ad 1`
	`[%{counter=2|counter=3}Story 2 and 3%]`

<%DesignClass%>	The design class of the story in the rendered context (frontpage or section design class).
<%ExternalLinks%>	A list of external links connected to the story. Output is generated through the following object files: FreestyleStoryItemExternalLinks.pbo and FreestyleStoryItemExternalLinkItem.pbo.
<%ImageGUID%>	The GUID of the image in the database.
<%ImageHeight%>	The height of the image.
<%ImageLink%>	The link to the main story photo. You use the field like this: `[%{Imagelink}<img src="<%Imagelink%>&MaxW=120&MaxH=80" border=0 hspace=1 vspace=1 alt="" align="right">%]` The options for setting the maximum width and height of the image have also been added to the link. You can use either one or both.
<%ImageOrientation%>	The image orientation. The field has the value P for portrait and L for landscape. It is written: `<%ImageOrientation%>`
<%ImageWidth%>	The width of the image.
<%InternalLinks%>	A list of links to other objects (of type story, gallery, forum, netguest) the story has been connected to. Output is generated through the following object files: FreestyleStoryItemInternalLinks.pbo, FreestyleStoryItemConnectionItem.pbo, FreestyleStoryArticleConnectionItem.pbo, FreestyleStoryItemGalleryConnectionItem.pbo, FreestyleStoryItemForumConnectionItem.pbo.
<%ISOCreatedDateTime%>	The created date and time in ISO format (YYYYMMDDHHMMSS). This field can be formatted using object script's **$d** modifier.
<%ISOLastModifiedDateTime%>	The last modified date and time in ISO format (YYYYMMDDHHMMSS). This field can be formatted using object script's **$d** modifier.
<%ISOPubDateTime%>	The publication date and time in ISO format (YYYYMMDDHHMMSS). This field can be formatted using object script's **$d** modifier.
<%IsEven%>	Whether the **Counter** is even.
<%IsOdd%>	Whether the **Counter** is odd.

<%KeywordLinks%>	A list of links to other stories with the same keyword. Output is generated through the FreestyleStoryItemKeywordLinks.pbo and FreestyleStoryItemKeywordLinkItem.pbo object files.
<%Link%>	The URL to the story.
<%Location%>	The location string of the story.
<%MainCategoryDescription%>	The description field of the story's main category.
<%MainCategoryInfo%>	The information field of the story's main category.
<%MainCategoryLink%>	The link to the section front relating to the story's main category.
<%MainCategoryName%>	The display name of the story's main category.
<%ObjectClass%>	Holds the objectclass parameter of the tag.
<%Priority%>	The priority given to the story.
<%ProfileID%>	The profile ID of the context in which the storylist is shown.
<%ProfileName%>	The name of the profile if profiles are used.
<%Rating%>	The rating of the story.
<%SectionCat%>	The ID for the category in which the story is shown. SectionCat is a flexible way to specify which template the page should use (without affecting the articles you get from the database).
<%SEOFilename%>	The SEO friendly filename specified for the image if any is set.
<%StoryCategory%>	The category ID for the category in which the story is shown.
<%StoryDate%>	The date the story was last modified. This field uses the **Dateformat** option in the tag for formatting.
<%StoryGUID%>	The story's GUID in the database.
<%StoryID%>	The article ID, which can be used with category and date as here:`http://www.yoursite.com/apps/pbcs.dll/article?AID=/20060721/PCSPORT/607210374`
<%StoryRef%>	Contains the story reference, which differs from <%ImageRef%> when using teasers
<%StorySite%>	Contains the site code of the story.
<%SubHeading%>	The subheading of the story.
<%Summary%>	This is the summary of the story. If the story does not have a summary, the first paragraph will be used.
<%SuperHeading%>	The superheading/kicker of the story.
<%TargetCategory%>	The target category of the profile **ProfileID**.
<%TeaserUrlTarget %>	Contains target on teaser url.
<%Theme%>	Outputs the theme field for the story.
<%ThemeLinks%>	A list of links to other stories with the same theme. Output is generated through the FreestyleStoryItemThemeLinks.pbo and FreestyleStoryItemThemeLinkItem.pbo object files.
<%Title%>	The story title.
<%Flow%> **From version 6.8.0sr11**	Contains the flow ID.
<%CurrentFlowStatus%> **From version 6.8.0sr11**	Contains the current status of the story within a flow using an ID.
<%CurrentSingleSale%> **From version 6.8.0sr11**	Contains the single sale Status ID associated to a story.

FreestyleStoryItemByline.pbo

The following variables are available in the object file. They are used to produce output for the **<%Byline%>** variable in FreestyleStoryItem.pbo w hen using smart byline, one iteration for each staff member associated with the displayed item.

<%Byline%>	Outputs a predefined byline with the author's name with a mailto link.
<%BylineNamePart%>	Contains the author's name as registered in the staff register.
<%Count%>	The total number of staff members in the list.
<%Counter%>	The staff member's number in the list.
<%Email%>	The staff member's email address.
<%Field_xxx%> From version 6.9.0sr2	Extracts extra fields for the staff member. Replace **xxx** with the name of the field. E.g., `<%Field_Facebook%>`
<%Firstname%>	Contains the author's first name as saved in the staff register.
<%ImgLink%>	Contains the URL of the author's picture.
<%StaffID%>	Contains the value from the staff ID field. This allows you to link to the information page for an author. The field is used like this: `<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>"><%BylineName Part%></a>` Please see pbs staffinfo for more information.
<%Surname%>	Contains the author's surname as saved in the staff register.

FreestyleStoryItemConnectionItem.pbo

Used with the field

```
<%ConnectionItems%>
```

in FreestyleStoryItemInternalLinks.pbo.

<%Date%>	The date in localized format.
<%Heading%>	The heading for the object.
<%IsForum%>	This field has the value 1 if the link is to a forum.
<%IsGallery%>	This field has the value 1 if the link is to a photo gallery.
<%ISODate%>	The date in ISO format.
<%Leadin%>	The lead-in for the object.
<%Link%>	The link to the object.
<%Category%> **From version 7.2.0sr18**	If object is a gallery or a story this variable holds its category.
<%ID%> **From version 7.2.0sr18**	If object is a gallery or a story this variable holds its ID.

<%Ref%> **From version 7.2.0sr18**	If object is a gallery or a story this variable holds its Ref.

FreestyleStoryItemExternalLinkItem.pbo

Used with the field

```
<%StoryListItemExternalLinkItem%>
```

in FreestyleStoryItemExternalLinks.pbo.

<%ImageLink%>	Contains the link to the photo defined in Story External Links in the Edit interface.
<%LinkDescription%>	The description of the link.
<%LinkID%>	The ID of the link.
<%LinkMainCategory%>	The main category for the link.
<%LinkRegion%>	The region for the link.
<%LinkSubCategory%>	The subcategory for the link.
<%LinkSubRegion%>	The subregion for the link.
<%LinkTitle%>	The title of the link.
<%LinkURl%>	The link's URL.
<%LogoLink%>	Contains the link to the logo defined in Story External Links in the Edit interface.

FreestyleStoryItemExternalLinks.pbo

Used to show links to external objects that are linked to the story.

<%StoryListItemExternalLinkItem%>	Shows external links.

FreestyleStoryItemForumConnectionItem.pbo

Used with the field `<%ForumConnectionItems%>` in StoryListItemInternalLinks.pbo.

The file uses the same fields as StoryListItemConnectionItem.pbo except `<%IS....%>` fields are not available.

<%ObjectClass%>	If you need to use the current object class with the forum template, add it to the link like this: `<%Link%>&ObjectClass=<%ObjectClass%>`

FreestyleStoryItemGalleryConnectionItem.pbo

Used with the field `<%GalleryConnectionItems%>` in [FreestyleStoryItemInternalLinks.pbo].

The file uses the same fields as [FreestyleStoryItemConnectionItem.pbo] except `<%IS....%>` fields are not available.

FreestyleStoryItemInternalLinks.pbo

Used to show links to other internal objects that are linked to the story.

<%ArticleConnectionitems%>	Shows links to connected stories.
<%Connectionitems%>	Shows internal links to both stories and photo galleries.
<%ForumConnectionitems%>	Shows links to connected forums.
<%GalleryConnectionitems%>	Shows links to connected galleries.

FreestyleStoryItemKeywordLinkItem.pbo

Please see KeywordLinkItem.pbo for information about available fields.

FreestyleStoryItemKeywordLinks.pbo

Please see KeywordLinks.pbo for information about available fields.

FreestyleStoryItemThemeLinkItem.pbo

Please see ThemeLinkItem.pbo for information about available fields.

FreestyleStoryItemThemeLinks.pbo

Please see ThemeLinks.pbo for information about available fields.

FreestyleStoryList.pbo

This file contains the main framework for the tag. It is possible to use the condition tags to exclude the whole file from being shown.

<%Items%>	This tag outputs all the items. The output is generated through the FreestyleStoryItem.pbo object file.
<%ItemCount%>	The number of items in the list.

pbs history

<pbs:history>

The history tag is used to list older stories in the category being viewed. It can be used both on the section front and in a story. The function this tag provides has been superseded by the Newslist tag. It is only documented for historical purposes. The {{<pbs:history> }}tag might be used as follows:

```
<pbs:history useobjects=1 dateformat="mm/dd/yyyy">
```

Options

Bgcolor	Sets the background color on the title, when an object file is not used.
Bredde	Sets the width of the output in pixels or percentage when object files are not used.
Category	Determines the category from which the history is to be retrieved. By default, the current category is used. Select one of the following if another category option should be used: • **Current** - Stories from the current category and subcategories. • **Category** - Stories from a single category. • **Category%** - Stories from a single category with subcategories. • **%** - All categories.
Color	Sets the color on the title above the links, if an object file is not used.
Count	Determines how many stories the history tag should display. By default, the value is set in the category register. Enter `Count=#` to override the setting in the category register.

Dateformat	Controls how the date will be shown by the <%\date%> tag. Note that it is also possible to use the <%isodate%> variables and format dates yourself. The default date format is dd/mm. This option can use other standard date format options.
Fontsize	Sets the font size of text when object files are not used.
Ledetekst	Sets the title when an object file is not used.
Objectclass	Determines which object files the system should use. By default, this option is disabled. By setting `Objectclass=1` the system will look for object files named History1.pbo and HistoryItem1.pbo. By using other numbers, the system will look for object files named History#.pbo where # is the number.
Object files	Enter `Useobjects=1` to use object files. By default, object files are not used.
Useobjects	This option controls the use of object files. By default, this option is disabled. Enter {{Useobjects=1 }}to make the system look for object files.
***Visref ***	Determines how many characters of the summary the <%text%> variable should use. By default, it uses 0 characters. Enter `visref=x` where x is any number between 0 and 9999.

History.pbo

The wrapper for the objects.

<%Items%>	The position where the code for each item will be output.

HistoryItem.pbo

The file HistoryItem.pbo is output for each item in the history list.

<%\Date%>	The story's date. Formatting is controlled by the **Dateformat** option.
<%Link%>	The link to the story.
<%Text%>	The summary of the story. The amount of text shown is determined by the *Leadin *option.
<%\Title%>	The title of the story.

pbs keywordindex

<pbs:keywordindex>

The index tag is used to build keyword indexes. The stories must be tagged with keywords to appear in the index.The tag must be placed on the template **gen_idx.pbs**. Stories must be tagged with one or multiple keywords to appear in the index. To link to the page, you must link to* /apps/pbcs.dll/index*. To specify a category, add **?Category=Categoryid**.

Options

Current	The keyword index only show stories from the category currently being viewed. This requires that the category is specified in the URL.
Descending	This option reverses the sort order of the index. By default, the index is sorted ascending.
Includesubs	The keyword index shows stories from the category currently being viewed and subcategories.
Objectclass	Determines which object class will be used.

Range	Controls which items are output. Range is set by using the option like this:
	`Range="0-9"`
	will output all entries with numerical keywords.
	`Range="0-Z"`
	will output all entries.
	`Range="A-Z"`
	will output all entries with alphabetic keywords.
	`Range="A-A"`
	will output all entries with keywords that begin with A.
	The default is A to Z.

Useobjects	This tag will use object files if they exist. To turn off the use of objects, enter `Useobjects=0`

KeywordIndexHeader.pbo

<%Header%>	The header. The header is the first letter of the alphabet.
<%TitleList%>	This field outputs the items.

KeywordIndexLink.pbo

<%ArticleLink%>	The link to the story.
<%ArticleTitle%>	The title of the story.
<%ArticleDate%>	The date the story was published.
<%IsoArticleDate%>	The date in ISO format. This date can be formatted using the variable formatting options.

KeywordIndexTitle.pbo

<%LinkList%>	This field outputs the items.
<%\Title%>	The keyword.

pbs mostread

<pbs:mostread>

Used to list the most frequently read stories for a category. Please see LiveArtStatUpdate on how to enable the use of `<pbs:mostread>`.

> Mostread does not support cachetime=xx so do not insert ! in this tag. It will run uncached.

Templates
This tag can be used on all templates.

Options

All	Enter this option to omit the date range filter. `<pbs:mostread count=10 all>`
IncludeNodate	Set this parameter to 1 (it is default 0) to include undated stories. `<pbs:mostread count=10 all includenodate=1>`

Category	Determines the category for which stories will be listed. By default, stories are listed for the current category. To list popular stories for all categories, use the option
	`Category="%"`
	You can also specify a single category by using
	`Category="NEWS"`
	To list a category with subcategories, use the option like this:
	`category="NEWS%"`

Count	Determines how many stories will be shown. By default, 10 stories are shown. Enter `Count=X`, to show X stories.
CurrentDate	Lists the most read stories considering only today's statistics, where today is the site's current publication date. For example: `<pbs:mostread showref=0 count=10 CurrentDate category=%>`

Days	Limits the age of the stories shown. This option makes sure that only articles in the days range, will be showen, this is useful when you dont want old stories to show up in the tag. By default, stories published in the last 100 days will be shown. The value can be set from 1 to 365, does not work with **All** and specifying both options will make the tag ignore **All.** If days is used without any other date intervals the default date interval is yesterday.
LastMonth	Lists the most read stories from the last month. For example, if the current date is May 5th, this option will list the most read stories from April 1st to April 30th.Best practice: If you use LastMonth, do not use LastWeek.
LastWeek	Lists the most read stories from the last week (Monday to Sunday). Best practice: If you use LastWeek, do not use LastMonth.
Minread	Sets the minimum of hits a story must have before it will be listed. If it's set to 10, all articles with 11 and up will be listed. By default, a story must have 5 views before it is listed. If MostEmailedArticles=1 or MostPrintedArticles=1 is in use, minread=x will decide the minimum of printed articles or emailed articles instead of the article stats. Enter `minread=x` where x is any number between 1 and 99.
MostEmailedArticles	Tracks the number of times that a story has been e-mailed (Tip a Friend). This option lists the stories that have been e-mailed most times on a chosen date.Best practice: If you use MostPrintedArticles, do not use MostEmailedArticles.
MostPrintedArticles	Tracks the number of times that a story has been printed. Update of the number of stories printed requires use of the PrintArt.pbo or that an additional "print=1" parameter be appended to the URL of the "Print article..." link. This option lists the stories that have been printed the most on a chosen date. Best practice: If you use MostEmailedArticles, do not use MostPrintedArticles.
Objectclass	Sets the object class for the tag. By default, object class is not used. Enter `objectclass=x` where x is any number between 1 and 999 to set the object class.
Profile	Displays the most read stories based on profile. By default, this option is disabled. Enter `Profile=X` to show stories with the profile X.
Relprofile	This function makes the system look for the most relevant profile for a story. If you specify the profile in the **MostRead** tag, the link to the story will contain the profile ID with the most matching points for the story. This function is used when your design is attached to the profile and not the category, or you have a **MostRead** tag on the story page showing stories from the same profile.
Showref	Determines how many characters of the summary will be shown. By default, no characters from the summary will be shown. Enter `Showref=150` to show the first 150 characters of the summary.

Site	Shows the most popular stories from another site. By default, this option is disabled. Enter {{Site=X }}in the tag to show stories from the site with site code X.
StatDays	Limits the age of the stories shown. StatDays uses the date for the last time the story was read which is the last time statistics changed on the story.For example, a story about Tasmania was published 100 days ago. This story might be read many times today, if a girl from Tasmania was married to a Danish prince yesterday. This story would not be shown with the Days option set to 7, but it would be counted if StatDays was set to 7. Please see **Days**.

MostRead.pbo

<%ItemCount%>	The number of stories.
<%Items%>	Lists the items.

MostReadItem.pbo

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of MostReadItemAccessControls.pbo
<%Counter%>	The position number of the story.
<%Category%>	The category of the story.
<%Date%>	The date for the story.
<%ExtLink%>	The link to the story when you use the site option.
<%Heading%>	The title of the story. This is an alias of **<%Title%>**
<%IsLastItem%>	This field has the value 1 when the last story in the list is shown.
<%ISODate%>	The date in ISO format.
<%Leadin%>	The summary for the story. This field only has a value if showref is larger than zero.
<%Link%>	The link to the story. From version 7.1.0sr15, 7.2.0sr8: If there are a SEO label associated with the story and the feature otherwise is enabled, the link will be SEO friendly.
<%MostReadStat%>	The total number of views for the story.
<%Place%>	The position number for the story with trailing period.
<%Position%>	The position number for the story with trailing period.
<%Title%>	The title of the story. We recommend using this variable because it contains the first nonempty of the followiing properties of the story: the Teaser Heading, Net Title or Main Title.
<%MainTitle%> **From version 6.9.0sr2**	The main heading of the story.
<%NetTitle%> **From version 6.9.0sr2**	The Net Title of the story.
<%SEOLabel%> **From version 7.1.0sr15, 7.2.0sr8**	The SEO label of the story.
<%SuperHeading%>	The super heading (Kicker) of the story.
<%TaxonomyList%> **From versions 6.6.0sr47, 6.8.0sr19, 6.9.0sr1**	A list of taxonomy words on the item. This is rendered by MostReadItemTaxonomy.pbo
<%TeaserHeading%> **From version 6.9.0sr2**	The Teaser Heading of the story.

<%AccessControlIdentifier%>	Indicates the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusId%>	Represents the current status of the story within a flow using an ID. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%>	Represents the current status of the story within a flow using text. This variable is used in conjunction with Access Control.
<%SingleSaleId%>	This field contains the text to output. For example, *You must be a subscriber to access this story. To buy a subscription...*

MostReadItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of MostReadItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

MostReadItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

MostReadItemTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

MostReadItemTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses MostReadTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

MostReadItemTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

pbs newslist

<pbs:newslist>

The **Newslist** tag is used to build overviews of stories. This tag can be used on all templates.

Option

AccessDomain	Used in ClickShare for setting the common domain for clicks.
Cachetime	The option cachetime is used to determine how long a Newslist should be cached. The function is typically used on front pages and for breaking news lists.By default, cachetime is disabled. To use the cachetime option, you must run the tag uncached. The option is used like this: `Cachetime=X` where X is the number of minutes the list should be cached.
Noduplicates	

This functionality allows a synchronization of <pbs:newslist> tags. By incorporating the NoDuplicates and SynchID parameters, the developer of a website is able to control whether or not stories are allowed to repeat, relative to two or more <pbs:newslist> tags. This enables easy removal of duplicate teasers in newslists. Note that <pbs:newslist> tags which you would like to be synchronized must be placed in the same file. By default, this attribute is disabled. Enter noduplicates=1, for the system to maintain a list of stories to be displayed. Setting the synch ID value to a specified number effectively groups all <pbs:newslist> tags that have the same synchronization ID. The following code provides an example of how to use the SynchID and NoDuplicates parameters:<pbs:newslist days=20 multisite=1 noduplicates=1 site="ap,cj" synchid=1 count=4 objectclass=1><pbs:newslist days=20 multisite=1 noduplicates=1 site="ap,cj" synchid=1 count=2 objectclass=1>

In the example above, both <pbs:newslist> tags share a common **SynchID** value, which allows SAXOTECH Online to group these tags together. By setting the **NoDuplicates** parameter, the system will potentially incorporate the results of each <pbs:newslist> into its list of stories. If a story is found to be repeated, the system will not add it to the list. As a result, it will present a list of non-repeating stories. Pre version 6.6.0sr28 the duplicate handling was done by means of a few settings, please see PBCSDupArtFilter.
When using NoDuplicates and the same SynchID on several newslists in the same file you have to make sure that unique cachefiles are generated

Example:

```
<pbs:!newslist cachetime=2
category="NEWS" count=3 days=5
useobjects=1 objectclass=1
noduplicates=1 synchid=1>
<pbs:!newslist cachetime=2
category="NEWS" count=3 days=5
useobjects=1 objectclass=1
noduplicates=1 synchid=1>
```

These two newslist tags are 100% identical, this will generate the same cachefile for the two tags. Effectively this means that the second newslist tag overwrites the cachefile from the first tag as the system builds cachefiles based on the parameters in a newslist tag.

What you should do to prevent this from happening is the following:

```
<pbs:!newslist cachetime=2
category="NEWS" count=3 days=5
useobjects=1 objectclass=1
noduplicates=1 synchid=1
uniquecache=1>
<pbs:!newslist cachetime=2
category="NEWS" count=3 days=5
useobjects=1 objectclass=1
noduplicates=1 synchid=1
uniquecache=2>
```

	A parameter, `uniquecache=X` will make the cache files be different, and the content will be correct in those cases. So a new parameter "uniquecache" has to be added to the tag with a unique value if they would otherwise be identical. We strongly recommend that you always add that parameter when using noduplicates and synchid, no matter if they are the same, or different.
Objectclass	Controls which object file will be used. Enter `Objectclass=1` to make the system look for News1.pbo and NewsItem1.pbo. The rules defining which object files will be used are described in the section About Object Files. You should also read the section on the folders in which the system will look for object files, which can be found in Directory structure.
SynchID	Setting this value to a specified number effectively groups all `<pbs:newslist>` tags that have the same synchronization ID. See **NoDuplicates**.
Useobjects	Determines whether object files should be used. If you enter `Useobjects=1`, the system will look for an object file to define the layout. The object files are described later in this chapter.
UseEmbeddedNewslist	UseEmbeddedNewslist="1"

Filter options

The newslist tag has the following output options. These options control which stories will be shown by the tag.

It is possible to display every tag parameter in any Newslist object file by using the following syntax:

```
<%TagParamXXXX%> where XXXX is the name of a Newslist tag parameter
```

you can also list all tag parameters by using the following variable:

```
<%AllTagParameters%> Returns all tag parameter with the corresponding values
```

Categories	When you intend to include all categories, do not use Categories=% or Category=%. If neither are provided, then all categories will automatically be included. Used when you want to select multiple categories for the Newslist and it is not possible to use the pattern matching that the category option supports. The category option is faster and should be used if possible. This option is only used when the option category cannot be used.To select multiple categories, enter them into the tag. For example,

```
Categories="NEWS,SPORTS"
```

selects stories from both the *News* and *Sports* categories as in this example:

```
Categories="news%,sports0[1-3],arts[^4],opinion"
```

The options are:

```
Categories="news%,sports%"
```

Selects stories from all News and Sports categories

```
Categories="news0[5-6],sports0[1-2]"
```

Selects stories from the categories news05, news06, sports01 and sports02. As with the category option, the range used must be one character. Writing

```
Categories="CATEGORYID[05-06]
```

is not supported.

```
Categories="news0[^5],sport0[^1]"
```

Selects stories from all categories except for news05 and sports01.

```
Categories="#category#"
```

Get current category from the URL

ExcludeCategories	Used when you want to exclude categories for the Newslist. To exclude multiple categories, enter them into the tag. For example,
	
	`ExcludeCategories="NEWS,SPORTS"`
	
	excludes stories from both the *News* and *Sports* categories as in this example:
	
	`ExcludeCategories="news%,sports0[1-3],arts[^4],opinion"`
	
	The options are:
	
	`ExcludeCategories="news%,sports%"`
	
	Excludes stories from all News and Sports categories
	
	`ExcludeCategories="news0[5-6],sports0[1-2]"`
	
	Excludes stories from the categories news05, news06, sports01 and sports02. As with the category option, the range used must be one character. Writing `ExcludeCategories="CATEGORYID05-06"` is not supported.

Channel	This option uses the **Channel** property from the editing interface to control which stories the tag should list. This option is typically used when you build a page that is to be used for either PDA or WAP distribution.The default is to output stories through the Web channel. The options for channel are: Web, PDA or WAP.The tag would appear as follows: {{<pbs:newslist channel=PDA> }}and the resulting output would be that only stoies with a matching channel will be output when this option is used.
Count	This option controls how many stories should be shown. The default is 5 stories. Enter from 1 to 999 stories to change the option. When you use profiles, the cache rows option on the profile will set the hard limit for how many stories the tag can show.
CounterStart	Determines the counter parameter's offset. Enter counterstart=# where # is the number used for the offset from the counter setting.

Createdby	Used when you want to select stories from specific users.
	
	Example:
	
	`<pbs:newslist createdby="UserId-3125617,UserId-3125122,UserId-111111">`
	
	The tag will select only stories submitted by those users.
	
	Example:
	
	`<pbs:newslist createdby="#currentuser#">`
	
	The tag will retrieve current logged in user id from the session object and list stories.

Date	This function allows you to pick stories from one date. By default, this is not used. By setting Date="99999999" you can select stories without a date. You can use any date in YYYYMMDD format to select stories from only that date. Example: `<pbs:newslist date="20110203">`
Days	Controls how many days back the system will look for stories. The default is 7 days. Enter from 1 to 999 days to change the option. For the sake of speed, you should never use more days than necessary. Setting this option to a high number will slow down the page.When you use profiles, the Week span option on the profile will set the hard limit for how far back the system looks for stories. If you have a week span of 7, Days=49 is the highest number of days that will have any meaning.
Daystart	This option controls how many days the tag should skip relative to today's date before it starts displaying stories. Setting Daystart=1 and Days=2 lists stories from yesterday. You cannot modify the date using Allow Date Override (Ado=1).
StartDate	This option lists stories from specific start and end date. It is use with option EndDate, it can not be used without EndDate. It should be noted, that this feature respects the value given in MaxDaysForNewslistDateRange (publicus.ini) and will change the start date accordingly if needed. This is working against the dato field, not publishFrom/To

EndDate	This option lists stories from specific start and end date. It is use with option StartDate, it can not be used without EndDate. It should be noted, that this feature respects the value given in MaxDaysForNewslistDateRange (publicus.ini) and will change the start date accordingly if needed. This is working against the dato field, not publishFrom/To. **Example:** `<pbs:newslist StartDate="20081201" EndDate="20090101" useobjects="1" objectclass="1">` The tag will select only stories between 20081201 and 20090101
Even	Enter `Even` in the tag to output only items with an even position code. By default this option is not used. Enter `Even` in the tag to output only items 2, 4, 6, 8, and so on.
ExactTaxonomyMatch	This settings controls whether the entire taxonomy tree is used when using Taxonomy words parameters. By default the entire taxonomy tree is used. This means that stories which have the taxonomy word identified in the newslist including the parent structure will be selected. If this settings is set to 1 then only stories which have specified taxonomy ids assigned to the directly will be selected.
ExternalLinksCount **From version 6.9.0sr18**	This option sets a max occurence of extrenal links on a news item. `<pbs:newslist externallinkscount="5" >`
FilmID	This option is like Location. It is used for selecting stories with a keyword of the **filmid** instead of location for Location.
IgnoreStatus	This option ignores published status on stories. `<pbs:newslist ignorestatus >`

InternalLinksCount **From version 6.9.0sr18**	This option sets a max occurence of internal links on a news item. `<pbs:newslist internallinkscount="5" >`	

Keywords	This option lets you select stories based on the keywords with which they are tagged. By default, this option is not used.You can use
	`keywords="politics,Cuba"`
	to select stories that have been tagged with both keywords. If your keyword has spaces, you must write the keyword with underscores. The option must be written like this:
	`keywords="keyword_with_spaces"`
	To select stories that have been tagged with the name of the category being viewed, use the keyword #category#. If you are viewing a subcategory and want to use the name of the main category as a keyword, use the keyword #maincategory#.	

Kwnot	This option separates the keywords with NOT. The default option for keywords is to use the AND separator. Entering `Kwnot=1` in the tag determines that the keywords will be separated with NOT.	
Kwor	This option separates the keywords with OR. The default option for keywords is to use the AND separator. Entering `Kwor=1` in the tag determines that the keywords will be separated with OR.	
Lastdate	Makes the tag only select stories from the last publication date. This will override the count and days options.	

Letter	Used with Alphasort to select which letters should be output. The system looks for stories with a Net title that matches the designated letter. If the story does not have a Net title, the story title will be used instead.This option has no default setting. There are several methods to use this option. • **Letter D** - Outputs all stories that start with the letter D. • **Letter=A-G** - Outputs all stories that start with letters from A to G. • **Letter=0-9** - Outputs all stories that start with numerals.
Location	Allows you to select stories based on the story's **Location** property. The text specified in this option must be an exact match for the value from the **Location** property on the story. By default, this option is not used. Enter `Location="XXX"` to make the tag select stories with XXX in the **Location** field.
Multisite	Allows the tag to retrieve stories from multiple site codes. Used with **Site. *Only use *Multisite** if articles are retrieved from multiple sites. The following code provides an example of how to use the Multisite parameter (note that articles are retrieved from two sites): `<pbs:newslist multisite=1 site="ap,cj">` By default this option is not used. If you enter Multisite in the tag, you can use the site option to select stories from multiple sites.
Odd	Enter `Odd` in the tag to output only items with an odd position code. By default this option is not used. Enter `Odd` in the tag to output only items 1, 3, 5, 7, and so on.
ParagraphCount	Used when you want to retrieve paragraph and paragraph images using newslist tag. `ParagraphCount="10"` will return the first 10 paragraphs of a story using NewsItemParagraphItem.pbo object file. The default is 0, meaning no paragraphs will be output.
Prio	Used to select stories on one priority only. Setting Prio=1 will select all stories with priority set to 1. To select stories without a defined priority, use the option Prio=0 in the tag. Prio will not work if using PrioSort and/or PrioOnly parameters.

Prioonly	Only stories with a defined priority will be shown. If you want to show stories without a defined priority, use the option Prio=0. If you use Prioonly, you must use Priosort too.
Priosort	Sorts the story list by priority. By default, this option is disabled and stories are sorted by timestamp. Enter {{Priosort=1 }}to sort stories by priority. The result is that stories are sorted by publication date then priority code.

Profile	Used to select stories based on profiles instead of categories. A profile is a collection of taxonomy words. For more information please see Taxonomy and profiles. By default,
	`<pbs:newslist profile>`
	takes the profile from the URL. For example, profile=1038. When you place a profile in the **Newslist** tag, the profile selected on the **Cat egory** settings page in the edit interface will be used. You can override this profile by entering
	`&Profile=XXX`
	in the URL, where XXX is the profile number. You can specify the profile directly by entering
	`Profile=XXX`
	in the tag. This will lock the tag to display stories matching profile number XXX. You can limit the selection of stories using categories with profiles. This means that stories matching a profile also will have to match the category.

ProfileLink	Used to turn off the profile in the link to the story. By default, the profile link is output. By setting {{Profilelink=0 }}in the tag, the profile will not be output.
ProfileSite	Determines from which site the profile should be selected. Enter `ProfileSite=xx` where xx is the site code for that site.

ProfileType	Determines the sub-profile to select. Enter `ProfileType=xxx` where xxx is the Profile Type ID.

Refant	Controls how many stories should have a lead-in. By default, all stories have a lead-in. Enter
	`refant=x`
	where x is any number from 1 to 99 to display lead-ins on the specified number of stories.Saxotech strongly recommends using objectscripting instead of the refant option to control how many stories should have a lead-in. By using objectscript, you can keep the number of object files low while still displaying a lead-in. For example:
	`NewsitemX.pbo [%{Counter=1|counter=2}<%summary%>%]`

Relprofile	This function makes the system look for the most relevant profile for a story. If you specify the profile in the Newslist tag, the link to the story will contain the profile ID with the most matching points for the story. This function is used when your design is attached to the profile and not the category, or you have a Newslist tag on the story page showing stories from the same profile.By default, the profile specified in the tag is used. By adding relprofile to the tag, the most relevant profile will be used in the link. **Relprofile logic** It uses the sorting functionality in this order: 1. Profile priority 2. Then Profile level (It takes the deepest profile in the profile tree) 3. Then Taxonomy level (It takes the deepest Taxonomy in the Taxonomy tree) Notice: that profile Id you are using in newslist tag, will not be taken in the relprofile logic. Example: If you have profile=1001in the tag, even if the profile 1001 has profile priority 1, it will not be a part of the matched profiles.

Reltaxonomy	This function enables you to show a list of related stories for each element in a newslist. This solution is based on taxonomy, where taxonomy is used to group stories so that they are shown together. The related stories will always be sorted by the Last Changed date. The function is used with the **RelStoriesItems.pbo** and **RelStoriesItem.pbo** object files.Note that this feature is hard on the database, so limited use is advised. The **Or** operator gives fewer database queries (because only one taxonomy word is used for the related taxonomy) so it should be used whenever possible.By default, the reltaxonomy option is disabled. Add Reltaxonomy=1 to the tag to enable the related taxonomy feature.
Reltaxonomycount	The number of related stories to show in the list under each element. By default, five stories are shown in each list. You can enter any number up to (2^31^)-1.
Reltaxonomyoperator	The **Or** operator gives fewer database queries (because only one taxonomy word is used for the related taxonomy) so it should be used whenever possible. By default a single taxonomy word is used for related taxonomy.To use reltaxonomyoperator enter Reltaxonomyoperator=And - Several taxonomy words are used for related taxonomy.
Reltaxonomytopnode	Must be set when using *Reltaxonomy. *If no topnode is wanted, this should be set to the root node (ID=0). This is not advised as the system will only match on the 5 taxonomy words with the lowest ID.Instead, a subtree should be created in the taxonomy tree, which is used to show the list of related stories. The first node of the subtree should be used as the Topnode. If the topnode is not set, the system will not show related stories and an exception will be raised. The option would appear as follows: `reltaxonomytopnode=[TaxonomyID]`
ShowBloggerInfo	Determines if blogger fields such as **CreatedByUserFirstName** is to be outputted (=1) or not(=0). Default is 0.
Site	Determines the sites from which the tag can retrieve stories. This option is used with *multisite.*By default, this option is not used. If you enter `Site="TB,SB,GJ"` in the tag, you can select stories from those three site codes.

Staff	Used when you want to select stories from a specific staff member.
	
	Example:
	
	`<pbs:newslist staff="b.jones">`
	
	The tag will select only stories with a smartbyline association to the staff member with ID "b.jones".
	The system will allow the query parameter "staff" to override the tag's parameter, e.g.,
	
	`/section?category=staffprofile&staff=b.jones`

Start	This option controls how many stories the tag should skip before it displays stories. This option can be used when you want to place an ad in the listing of stories or use multiple Newslist tags on the same page to control layout.By default the system starts at the first story but you can set the option to start at any story number between 1 and 999.
Status	This option makes it possible to show stories depending on their status. To enable this, it must be activated in the Publicus.ini by setting the following: `AllowChangingPublishedStatusOnNewslistTags=1` in the GENERAL section of the publicus.ini. This enables the status parameter on the Newslist tag. This parameter must be set to a set of valid statuses separated by comma. The valid statuses are: • 0: Pending • 1: Unpublished • 7: Restricted • 8: Archived • 9: Published • 99: Backed up • 10: Locked For example: `<pbs:newslist days=2 status="1,9,3,4">` It is important to use quotes (") around the status string. Otherwise, it only uses the first element in the list. Use caution when using the category or profile from the URL in the tag (#category#) because this may allow readers to find unpublished stories from other categories and/or profiles.
Taxincl or Taxinclude	This option is used the same way as **Profile**. See **Profile** for more information.

Taxonomywords	You can output stories matching specific taxonomy words. By default, this option is not used. Use `Taxonomywords="12,14"` in the Newslist tag. When you specify multiple words, they will be separated by AND. By adding `Kwor=1` to the tag, the terms will be separated by OR.Example:`<pbs:newslist taxonomywords="12,14" kwor=1>`. Please read ExactTaxonomyMatch.
ExcludeTaxonomyWords From version 7.1.0sr13 and 7.2.0sr2	This is used to exclude stories based on them having one or more of the taxonomy ids given here directly attached to the story. The input is a comma separated list of taxonomy ids.
ExcludeTaxonomyBranches From version 7.1.0sr13 and 7.2.0sr2	This is used to exclude stories based on them having one or more taxonomy words in the branches below the ids given here. The input is a comma separated list of taxonomy ids.
Themes	Allows you select stories based on theme. By default this option is not used. Using this option allows you to select stories using multiple themes. The option is used like this: `themes="computers,telephone,internet"`

Sort options

By not using any sort options, stories will be sorted this way:

- The default sort order when profiles are not used is date, modified time, Category ID and Article ID with lowest Article ID first.
- The default sort order when profiles are used is date, priority and Article ID with highest Article ID first.
- When you combine the sort options, Alphasort will take precedence over Priosort and Datesort. Datesort takes precedence over Priosort when these two options are used together.
- The following options for sort are available:

Alphasort	Sorts the stories alphabetically. Only the first letter of the title will be used. By default, this option is not used. Enter `Alphasort=1` to sort the stories alphabetically by the first letter of the title.
AvgRankingsort	Sorts the stories by average ranking. By default this option is not used. `avgrankingsort=1`

Datesort	Sorts the stories by the following criteria in the listed order: • By the Publication date or the Publish Control From date (whichever is most recent) with the newest story first. • By either the Last Modified time or the Publish Control From time (whichever is most recent)Using Datesort disables Priosort. By default, this option is not used. Enter `Datesort=1` in the tag to sort the stories by publication date.

Extrasortfield	When **ExtraFieldDef** is used in Publicus.ini, the Newslist tag can be sorted by the extra fields set up in that section. Please see Extra fields definition. By default, this option is not used.
	`ExtraSortField=1`
	enables sorting by extra fields. The sorting is done on the FIRST field in the fields parameter. The fields are taken from the fields defined for EditExtraFields in the Extra fields definition section in the **Publicus.ini**. For example:
	`<pbs:newslist category="MOVIE" count=10 useobjects=1 objectclass=1 profilelink=0 ExtraSortField=1 ExtraSortFieldOrder=asc fields=movie_titel,movie_manuscript>`

Extrasortfieldorder	Sets up a sort order for the ExtraSortField. Legal values are **asc** and **desc**. By default, the sort order is ascending. `ExtraSortFieldOrder=desc` sets up a descending sort order for the ExtraSortField.

Letter	Used with Alphasort to select which letters should be output. The system looks for stories with a Net title that matches the designated letter. If the story does not have a Net title, the story title will be used instead. By default this option is not used. Entering
	`Letter=D`
	will output all stories that start with the letter D. Entering
	`Letter=A-G`
	will output all stories that start with letters from A to G. Entering
	`Letter=0-9`
	will output all stories that start with numerals.

Priosort	Sorts the stories by the following criteria in the listed order: 1. By either the Publication date or the Publish Control From date (whichever is most recent) 2. By priority 3. By either the Last Modified time or the Publish Control From time (whichever is most recent)Web guests will see today's prio1-n stories, yesterdays prio1-n stories, etcetera. If you want to sort stories by priority and then by date, use **Priosortonly**. By default this option is not used. Enter `Priosort=1` to sort the stories by date and then by priority.
Priosortonly	Sorts the stories by the following criteria in the listed order: 1. By priority 2. By either the Publication date or the Publish Control From date (whichever is most recent) 3. By either the Last Modified time or the Publish Control From time (whichever is most recent) Example: `<pbs:newslist priosortonly>` If you use priosortonly, you must use Priosort too.
Ratingsort	Sorts the stories by rating. By default this option is not used. Enter `Ratingsort=1` to sort the stories by rating.
Sort From versions 7.0.0sr17 and 7.1.0sr6	A sorting parameter that allows for detailed control of what to sort on and in what direction to sort. For now the following fields can be specified: • Created - sorts on the story's create time (same as the <%ISOCreateDateTime%> variable in newsitem.pbo) • Date - sorts on the story's date (same as the <%StoryDate%> variable in newsitem.pbo) • Heading - sorts on the story's heading (as per the basic heading in the Edit UI) • Modified - sorts on the story's last modified time (same as the <%ISOLastModifiedDateTime%> variable in newsitem.pbo) • Priority - sorts on the story's priority (same as the <%Prio%> variable in newsitem.pbo) • Title - sorts on the story's teaser title, net title, and story title whichever the system finds to have value first (same as the <%Title%> variable in newsitem.pbo similar to **alphasort**) The sort order is specified by giving a field followed by a direction in a comma separated list, so if for example if you want to order by priority and then last modified, you would use: `<pbo:newslist ... sort="priority asc, modified desc" />`

Formatting options

These options control some of the formatting of the content output by the system and determine whether related content should be shown.

Alltext	Makes the system output all the text in the first paragraph. The variable `<%text%>` is used for displaying the story.
Dateformat	Sets the format for the preformatted date field. The default setting for this option is language dependent. The system will output the dates in the normal format for your language.Format the dates by using `Date format="mmmm dd, yyyy"` or any other date format option. You can also format the date field directly in the object file.Please see For matting dates for more information.
ExternalLinks	Used to enable links to external stories related to the story currently being viewed.By default, this option is not used. Enter `ExternalLinks=1` to show links to related stories outside your website.With the parameter enabled the system will look for and act on the <%ExternalLinks%> tag in the NewsItem.pbo. That tag calls the NewsItemExternalLinks.pbo file. The NewsItemExternalLinks.pbo will recognize the tag <%NewsItemExternalLinkItem%> that uses the pbo by the same name.

Extrapictures
From version 6.6.0sr20
	Controls output of the extra pictures on the story. The default is
	`Extrapictures=0.`
	Enter
	`Extrapictures=1`
	to enable output of extra pictures. Enter
	`Extrapictures=0`
	to disable output of extra pictures.

Fields	Specifies which extra fields should be available in the Newslist. The fields are specified in a comma separated list like this: `Fields="Edition,Section,Tearsheet"`
Image	Determines whether images will be shown in the Newslist. Enter Image=1 to show images in the newslist.
InternalLinks	Used to enable links that list stories related to the story currently being viewed. By default, this option is not used. Enter `InternalLinks=1` to show links to related stories.
KeywordLinks	Enables links to stories with the same keywords as the story currently being viewed. By default, this option is not used. Enter `KeywordLinks=1` to show links to stories with the same keywords.
KeywordLinksCount	This option tells the system how many stories with the same keywords should be listed. By default 5 stories are shown. Enter `KeywordLinksCount=2` to show two stories, for example.

KeywordLinksMonths	This option controls how many months back the system should look for stories with the same theme. By default the system looks back 3 months. Enter `KeywordLinksMonths=6` if the system should look back 6 months, for example.
Leadin	Controls how many characters of the lead-in should be displayed. If there is no lead-in, the first paragraph will be used. By default there is no lead-in.Enter any number between -1 and 9999 to change the number of characters displayed in the lead-in. Entering -1 will result in the system displaying the entire first paragraph of the story.
Multimedia	Enables direct links to multimedia objects connected to a story. By default Multimedia links are not output. Enter `Multimedia=1`to show links to multimedia objects when the field `<%MultiMediaItem%>` is present in the object file.
ProfileHeader	Determines whether a profile header will be shown for the newslist. Enter `ProfileHeader=1` to show a profile header for the newslist.
Showref	Determines how many characters of the summary will be shown. By default, no characters from the summary are shown. Enter Showref=# where # is the number of characters from the summary to be shown.
Source	Determines whether the source will be displayed in the newslist. Enter `Source=1` to display source in the newslist.
Teaser	This option is used for turning off the use of teasers on stories. By default, the teser registered on the story will be used. Enter `Teaser=0` to disable the teaser. The standard title and summary will be used.
ThemeLinks	Enables links to stories with the same theme as the story currently being viewed. By default, this option is not used. Enter `Themelinks=1` to show links to stories with the same theme.
ThemeLinksCount	This option tells the system how many stories with the same theme should be listed. By default, 5 stories are shown. Enter `Themelinks count=2` to show two stories, for example.
ThemeLinksMonths	This option controls how many months back the system should look for stories with the same theme. By default, the system looks back 3 months. Enter `ThemeLinksMonths=6` if the system should look back 6 months, for example.
Timeformat	Sets the format for the `<%\date%>` parameter in the object file. You can also format the time field directly in the object file. By default, time is displayed as hh:mm using a 24 hour clock. Please see Formatting dates for more options.

Special options

Allow date override (Ado)	Used to allow the date to be overridden in the URL. This option must be set if you want to use the **Frontpagearchive** function. Ado is ***not*** t o be used with Daystart. By default, this option is not used. Enter `Ado=1` to override the date used by the Newslist tag.
Frontpageflag	Makes the Newslist tag on the front page link to the template **gen_far t.pbs** instead of **gen_art.pbs**. Please see Template mapping. By default, this option is not used. Enter `Frontpageflag=1` to use gen_fart.pbs.
Frontpageflagforce	This makes the Newslist tag on all templates link to the template **gen _fart.pbs** instead of **gen_art.pbs**. By default, this option is not used. Enter `FrontpageflagForce=1` in the tag to use gen_fart.pbs.
GeoData	Used when you want to retrieve Geo Data using newslist tag. `GeoData="3"` will return the first 3 geo locations using NewsItemGeoData.pbo, New sItemGeoDataItem.pbo object files.
showCommentsCount	Used when you want to retrieve the number of user comments on a story using newslist tag. `ShowCommentsCount="1"`

There are multiple object files that can be used by the Newslist tag. Please see About Object Files and Position-based design. This section lists the base names.

From version 6.9.0sr18

News.pbo

This file contains the main framework for the tag. It is possible to use the condition tags to exclude the whole file from being shown.

<%Items%>	This tag outputs all the items.

NewsItem.pbo

The following list of parameters is available for use in the object file.

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of NewsItemAccessControls.pbo
<%AllTagParameters%>	Returns all tag parameters with the corresponding values.
<%ArticleStatus%>	Contains the status of the story. See Article.pbo for available status. This is only true when the 'ignorestatus' parameter is used on the newslist tag.
<%ArtID%>	The article ID for shorter links, which can be used without category and date, as here: `http://www.yoursite.com/apps/pbcs.dll/article?AID=9999607210374`. This field is used for cellphone payment for stories.
<%ArtIDLink%>	The complete story ID. It is for use with cellphone payment of stories.
<%AvgRanking%>	Contains the average ranking of a story.
<%BayesianRanking%>	Bayesian ranking of a story.
<%BlogEntry%>	Contains whether the story is a user generated story (blog entry) (=1) or not (=0)
<%Byline%>	The byline on the story.
<%CaptionBody%>	The main text for the picture.
<%CaptionByline%>	The byline for the photo.
<%CaptionHead%>	The heading for the picture.
<%Category%>	The category ID for the category in which the story is shown.
<%CategoryDescription%>	Current category description for an item.
<%CategoryInfo%>	Current category info text for an item.
<%CategoryLink%>	The link to the category. This field is often used in the following link: `<a href="<%categorylink%>"><%Categoryname%></a>`
<%CategoryName%>	The name of the category in which the story is placed.

<%Counter%>	This field starts with the value 1 and is increased by one for each story.
	
	`[%{counter=1}<b>Breaking News</b>%]`
	`[%{counter=1}Show Ad%]`
	`[%{counter=2|counter=3}Article 2 and 3%]`
	`[%{counter=3}Show Ad%]`
	`[%{counter>3}Article 4, 5, 6....%]`
	
	Melt your newslist tags together with this feature and decrease database connections.

<%CreatedByUserId%>	Contains the creator's user ID if the story is a blog entry.
<%CreatedByUserFirstName%>	Contains the creator's first name if the story is a blog entry.
<%CreatedByUserSurName%>	Contains the creator's last name if the story is a blog entry.
<%CreatedByUserEmailAdr%>	Contains the creator's email address if the story a blog entry.
<%CreatedByUserNickName%>	Contains the creator's nickname if the story a blog entry.
<%CurrentFlowStatus%>	Contains the current status of the story within a flow using an ID.
<%Date%>	The date the story was last modified. This field uses the **Dateformat** option in the tag for formatting.
<%DateHeader%>	This field outputs the date, but only when the date changes in the listing. This field uses the **Dateformat** option in the tag for formatting.
<%Dateline%>	The dateline from the story.
<%DesignClass%>	The value of the design class.
<%DesignClassChanged%>	The value 1 when the design class changes in the list of stories. It is written:`<%DesignClassChanged%>`
<%ExternalLinks%>	Used to show links to external objects that are linked to the story. For more information please see NewsItemExternalLinks.pbo
<%ExtraPictures%>	Extracts extra pictures attached to a story. These pictures must be imported from your editorial system. To use this variable you must have the object files NewsItemExtraPictures.pbo and NewsItemExtraPictureItem.pbo.
<%Field_xxx%>	This field is used for extracting extra fields from the story. Replace xxx with the name of the field. The fields must be specified in the tag. The maximum number of characters that will be output for extra fields is 255.
<%ForumLink%>	Used to link to the forum on the story.
<%GeoData%>	Contains story geo data.
<%Heading2%>	This is the subtitle for the story.
<%ID%>	The article ID, which can be used with category and date as here:`http://www.yoursite.com/apps/pbcs.dll/article?AID=/20060721/PCSPORT/607210374`
<%Identifier%>	Contains the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.
<%ImageHeight%>	The height of the image.

<%Imagelink%>	The link to the main story photo. You use the field like this: `[%{Imagelink}<img src="<%Imagelink%>&MaxW=120&MaxH=80" border=0 hspace=1 vspace=1 alt="" align="right">%]` The options for setting the maximum width and height of the image have also been added to the link. You can use either one or both.
<%ImageOrientation%>	The image orientation. The field has the value P for portrait and L for landscape. It is written: `<%ImageOrientation%>`
<%ImageRef%>	The reference to the image displayed. It is useful for zooming and required for zooming on teaser images.
<%ImageWidth%>	The width of the image.

Image zoom	The Newslist tag does not support the zoom function you have on the story page. It is possible to build this function using the link field. You do it one of these ways:
	
	`<a href="<%pbcs.dll%>/misc?url=/templates/zoom.pbs&<%Imagelink$c(23)%>">Zoom</a>`
	`<a href="<%pbcs.dll%>/misc?url=/templates/zoom.pbs&aid=<%ArtID%>">Zoom</a>`
	`<a href="<%pbcs.dll%>/misc?url=/templates/zoom.pbs&dato=<%IsoCreateDateTime$d("yyyymmdd")%>&kategori=<%Category%>&lopenr=<%id%>&ref=<%ImageRef%>">Zoom</a>`
	
	The latter zoom link is the only one that works with Teaser images.

<%Ingres%>	**This field outputs** the summary field. It will not show parts of the first paragraph if a summary does not exist. This field is used with the option **Alltext.**
<%InternalLinks%>	Used to show links to internal objects that are linked to the story. For more information please see NewsItemInternalLinks.pbo. InternalLinks=1 MUST be set in the pbs newslist tag for this variable to be available.
<%IsCurrent%>	This field has the value 1 if the story is currently being viewed. This function can only be used when you use the Newslist tag on a story page. This function can be used to make the current story bold in the list of stories.
<%IsLastItem%>	This field has the value 1 when the last story in the list is shown.

<%IsoCreateDateTime%>	The creation date and time in ISO format. (YYYYMMDDHHMMSS). For more information on formatting, please see <%ISOpubdate%>. This field can be formatted using object script. It is written: `<%IsoCreateDateTime%>`
<%IsoDateTime%>	The modification date and time in ISO format. (YYYYMMDDHHMMSS). For more information on formatting, please see <%ISOpubdate%>. This field can be formatted using object script.
<%IsoLastModifiedDateTime%>	The last modified date and time in ISO format. (YYYYMMDDHHMMSS).
<%IsoPubDate%>	The publication date in ISO format. (YYYYMMDD). This variable should be used in conjunction with date formatting functions. Please read the section Formatting dates. This field can be formatted using object script.
<%IsoPublishedDate%>	The publication date for the story in ISO format. `<%IsoPublishedDate%>` If publish control are activated and this publish control have a newer date then published date, will the value of publish control take over.
<%IsoPublishFromDateTime%>	The publishing date and time in ISO format. (YYYYMMDDHHMMSS). For more information on formatting, please see <%ISOpubdate%>. This field can be formatted using object script. It is written: `<%IsoPublishFromDateTime%>`
<%IsoPubDateTime%>	The publication date and time in ISO format (YYYYMMDDHHMMSS). This field can be formatted using object script. It is written: `<%IsoPubDateTime%>`
<%KeywordLinks%>	Outputs stories that have the same keywords as the current story. The function uses the object files NewsItemKeywordLinks.pbo and NewsItemKeywordLinkItem.pbo.
<%Link%>	The URL to the story.
<%MainCategoryLink%>	The link to the main category if you are listing an entry from a subcategory. This field is often used in the following link:{{<a href="<%MainCategorylink%>"><%MainCategoryname%>}}If the story is placed in the main category, this variable will contain no value.
<%MainCategoryDescription%>	The description of the main category for the category in which the story is placed if the story is placed in a subcategory.
<%MainCategoryInfo%>	The info text of the main category for the category in which the story is placed if the story is placed in a subcategory.

<%MainCategoryName%>	The name of the main category for the category in which the story is placed if the story is placed in a subcategory. If the story is placed in the main category, this variable will contain no value. It is written: `<%MainCategoryname%>`
<%MainProfile%>	Contains the profile chosen by the user in the edit interface as main profile when using multiprofile. the setting OnlyUseEmbeddedNewsList in publicus.ini must be set to 1.
<%MultiMediaItem%>	Used to link directly to a multimedia object that is attached to the story. It targets the object with direct link enabled. This field uses the object file NewsItemMultiMedia[Extension].pbo. [Extension] is replaced by the media type extension.
<%ObjectClass%>	Contains the value of objectclass from the newslist tag.
<%Paragraphs%>	Renders the content of NewsItemParagraphItem.pbo
<%Prio%>	The priority given to the story.
<%Profileid%>	The profile ID for the item.
<%ProfileName%>	The name of the profile when profiles are used.
<%PubDate%>	The publication date. This field uses the date **format** option in the tag for formatting.
<%PubDateTime%>	The publication date and time. This field uses the **Dateformat** and **Timeformat** options in the tag for formatting.
<%PublishedStatus%>	Same as ArticleStatus
<%Rating%>	Outputs the rating on the story. This feature is typically used in movie reviews. The field outputs just the rating value and it can be used to show an image like this: `[%{rating>0}<img src="/graphics/<%rating%>.gif">%]`
<%RelatedStories%>	Lists the stories related by taxonomy.

<%Sectioncat%>	The ID for the category in which the story is shown. SectionCat is a flexible way to specify which template the page should use (without affecting the articles you get from the database).
	• Example 1: You want all teasers in a newslist to use your SPORT article template (sport_art.pbs). In your NewsItem.pbo, enter:
	`<%Link%>&SectionCat=SPORT`
	The resulting URL in your newslist:`/apps/pbcs.dll/article?AID=/20060922/NEWS/60922080/2162&SectionCat=SPORT`**Example 2: Get a specific include file, based on the actual *SectionCat**parameter:
	`<!--#include virtual="/includes/#sectioncat#_header.inc"-->`
	If **SectionCat** does not exist in the URL, the system will look for {{/includes/default_header.inc.

<%SEOLabel%> **From version 7.0.0sr11**	Contains the SEO label of a story.
<%SingleSaleIdent%>	Contains the single sale Status ID associated to a story.
<%StoryClass%>	The value of the story class. This contains the one character value set in Publicus.ini, not the name shown in the edit interface.
<%StorySite%>	Contains the site code of the story.
<%SourceHostName%>	Outputs the host name (www.source.com) of the site from which the story is taken.
<%SourceName%>	The name of the site from which the story is retrieved. Used with multisite in the tag.
<%StoryCategory%>	Contains the category ID of the story.
<%StoryDate%>	Contains the date of the story.
<%StoryGuid%>	The GUID for the story.
<%StoryID%>	Contains the lopenr / StoryID of the story.
<%StoryProfileList%>	Lists all profiles in which the current story is listed. This variable will only be available when using the Multi Priority option. Please see MULTIPRIORITY section.
<%StoryRef%>	Contains the story reference, which differs from <%ImageRef%> when using teasers
<%Summary%>	This is the summary of the story. If the story does not have a summary, the first paragraph will be used. To avoid outputting the first paragraph of the story (if a summary does not exist) use `<%ingres%>`. For more information please see `<%ingres%>`.
<%Superheading%>	Outputs the super heading. You can use object scripts to force capitalization of the super heading. For example,<%Superheading$u%> will show as 'THIS IS A TEST'<%Superheading$p%> will show as 'This is a test'<%Superheading$n%> will show as 'This Is A Test'<%Superheading$l%> will show as 'this is a test'
<%TagParamXXXX%>	Outputs a parameter from a newslist tag, where XXXX is the name of a Newslist tag parameter.

<%Taxonomywords%>	Outputs the list of taxonomy words of the story. The function uses the object files NewsItemTaxonomyWords.pbo and NewsItemTaxonomy WordItem.pbo.
<%TaxonomyList%>	Outputs the taxonomy words attached to the story using NewsItemTaxonomy.pbo, NewsItemTaxonomyItem.pbo, and NewsItemTaxonomy ItemPathItem.pbo. It is not possible to Specify a delimiter (like in article.bpo)
<%Text%>	Outputs the first paragraph of the story. This field is used with the option **Alltext.**
<%TeaserUrlTarget %>	Contains target on teaser URL.
<%Theme%>	Outputs the theme field for the story.
<%ThemeLinks%>	Outputs stories that have the same theme as the current story. The function uses the object files NewsItemThemeLinks.pbo and NewsIte mThemeLinkItem.pbo.
<%Time%>	The time the story was last modified. This field uses the **Timeformat** option in the tag for formatting.
<%Title%>	The story title.
<%TopStory%>	Top story is set on all stories with a priority between 1 and 3. Top story can be used to set class on a tag, as in this example: `<a [%{TopStory} class="hot"%] href="<%Link%>"><%Heading%></a> ` `Class="hot"` is only output if the story is a Top story. If the story is not a Top story, no class tag will be output.
<%TotalNoofVotes%>	Contains the total number of votes on the story.
<%UserCommentsCount%>	Number of user comments on a story You need to have the following parameter enabled in order for this to work: `<pbs:newslist showCommentsCount=1>`
<%UserCommentsStatus%>	Contains the status of story user comment (0 off, 1 enabled, 2 locked) You need to have the following parameter enabled in order for this to work: `<pbs:newslist showCommentsCount=1>`
<%UserSubmitDeleteLink%>	Contains a direct link that deletes the story when visited (if the user is the story's creator).

NewsItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of NewsItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

NewsItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

NewsItemArticleConnectionItem.pbo

Used with the field `<%ArticleConnectionItems%>` in NewsItemInternalLinks.pbo.

The file uses the same fields as NewsItemConnectionItem.pbo except `<%IS...%>` fields are not available.

NewsItemByline.pbo

Same as ArticleByline.pbo with the following differences:

> This object file will only be used if Smart Byline are used on the specific article.

<%Address1%>	The first part of the address as defined in the staff registry. *Not available in NewsItemByline.pbo.
<%Address2%>	The second part of the address as defined in the staff registry. *Not available in NewsItemByline.pbo.
<%Address3%>	The third part of the address as defined in the staff registry. *Not available in NewsItemByline.pbo.
<%Address4%>	The fourth part of the address as defined in the staff registry. *Not available in NewsItemByline.pbo.
<%Byline%>	Outputs the names of all authors with the names as links to the e-mail addresses of those authors the system finds in the register. The function also outputs the other lines in the story.
<%BylineEmailPart%>	Contains the author's e-mail address as saved in the staff register. The same functions as for **<%BylineNamePart%>** are available here. *Not available in NewsItemByline.pbo.
<%BylineNamePart%>	Contains the author's name as registered in the staff register. If you have two names in the byline you also have the field **<%BylineNamePart2%>**. This function supports up to 9 names in the byline.
<%City%>	Used for Smart Bylines. Contains the author's city as saved in the staff register. *Not available in NewsItemByline.pbo.
<%CreatedByUserId%>	Contains the creator's user ID if the story is a blog entry.
<%CreatedByUserFirstName%>	Contains the creator's first name if the story is a blog entry.

<%CreatedByUserSurName%>	Contains the creator's last name if the story is a blog entry.
<%CreatedByUserEmailAdr%>	Contains the creator's email address if the story a blog entry.
<%CreatedByUserNickName%>	Contains the creator's nickname if the story a blog entry.
<%Department%>	Used for Smart Bylines. Contains the author's department as saved in the staff register. *Not available in NewsItemByline.pbo.
<%DesignClass%>	Contains the design class used.
<%Extension%>	Used for Smart Bylines. Contains the author's work phone number extension as saved in the staff register. *Not available in NewsItemByline.pbo.
<%Fax%>	Used for Smart Bylines. Contains the author's fax number as saved in the staff register. *Not available in NewsItemByline.pbo.
<%Firstname%>	Used for Smart Bylines. Contains the author's first name as saved in the staff register.
<%Fylkenr%>	Used for Smart Bylines. Contains the author's fylkenr as saved in the staff register.
<%Field_xxx%> **From version 6.9.0sr2**	Extracts extra fields for the staff member. Replace **xxx** with the name of the field. E.g., `<%Field_Facebook%>`
<%Group%>	Used for Smart Bylines. Contains the author's group as saved in the staff register. *Not available in NewsItemByline.pbo.
<%ImgLink%>	Contains the URL of the author's picture. For additional authors use the field **<%ImgLink2%>**. This tag supports up to 9 people.
<%Information%>	Used for Smart Bylines. Contains the author's information as saved in the staff register. *Not available in NewsItemByline.pbo.
<%IsPrimary%>	Used for Smart Bylines. `IsPrimary=1` indicates that the author is the primary author.
<%Kommunenr%>	Used for Smart Bylines. Contains the author's kommunenr as saved in the staff register.
<%LastName%>	Used for smart bylines. Contains the author's surname as saved in the staff register. *Not available in NewsItemByline.pbo.
<%Line1%>	Contains the first line of the byline field.
<%Line2%> etc...	You can increase the line number to output the different lines of the byline field.
<%MobilePhone%>	Used for Smart Bylines. Contains the author's mobile phone number as saved in the staff register. *Not available in NewsItemByline.pbo.
<%Name%>	Outputs the names of all authors without e-mail links. The function also outputs the other lines in the story. *Not available in NewsItemByline.pbo.

<%Namecount%>	Contains the number of authors the system has recognized. It can be used like this: `[%{namecount=1}<%BylineNamePart1%>%|%<%BylineNamePart%> and <%BylineNamePart2%>%]` *Not available in NewsItemByline.pbo.
<%Office%>	Used for Smart Bylines. Contains the author's office as saved in the staff register. *Not available in NewsItemByline.pbo.
<%PagerNumber%>	Used for Smart Bylines. Contains the author's pager number as saved in the staff register.
<%Position%>	Used for Smart Bylines. Contains the author's position as saved in the staff register. *Not available in NewsItemByline.pbo.
<%PrivateEmailAddress%>	Used for Smart Bylines. Contains the author's private email address as saved in the staff register. *Not available in NewsItemByline.pbo.
<%PrivatePhone%>	Used for Smart Bylines. Contains the author's private phone number as saved in the staff register. *Not available in NewsItemByline.pbo.
<%PrivateWebAdress%>	Used for Smart Bylines. Contains the author's private web address as saved in the staff register. *Not available in NewsItemByline.pbo.
<%StaffID%>	Contains the value from the staff ID field. This allows you to link to the information page for one author. The field is used like this: `<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>"><%BylineName Part%></a>` Please see pbs staffinfo for more information.
<%Surname%>	Used for Smart Bylines. Contains the author's surname as saved in the staff register.
<%WorkEmailAddress%>	Used for Smart Bylines. Contains the author's work email address as saved in the staff register. *Not available in NewsItemByline.pbo.
<%WorkPhone%>	Used for Smart Bylines. Contains the author's work phone number as saved in the staff register. *Not available in NewsItemByline.pbo.
<%WorkWebAddress%>	Used for Smart Bylines. Contains the author's work web address as saved in the staff register. *Not available in NewsItemByline.pbo.
<%Zipcode%>	Used for Smart Bylines. Contains the author's zip code as saved in the staff register. *Not available in NewsItemByline.pbo.

NewsItemConnectionItem.pbo

Used with the field

```
<%ConnectionItems%>
```

in NewsItemInternalLinks.pbo.

<%Counter%>	This field starts with the value 1 and is increased by one for each link.
<%Date%>	The date in localized format.

<%Heading%>	The heading for the object.
<%IsForum%>	This field has the value 1 if the link is to a forum.
<%IsGallery%>	This field has the value 1 if the link is to a photo gallery.
<%IsNetGuest%>	This field has the value 1 if the link is to a NetGuest.
<%ISODate%>	The date in ISO format.
<%Leadin%>	The lead-in for the object. The leadin length is depending on the ShowRef parameter on the newslist tag. Default is 0.
<%Link%>	The link to the object.
<%ObjectClass%>	Output the ObjectClass
<%TotalCount%>	Contains the total number of items returned.
<%TeaserUrl%> **From version 6.9.0sr29, 7.0.0sr4**	The External URL of a teaser link, if available. Example: <a href="[%{TeaserUrl}<%TeaserUrl%>%|%<%Link%>%]"><%Heading%>
<%Category%> **From version 7.2.0sr18**	If object is a gallery or a story this variable holds its category.
<%ID%> **From version 7.2.0sr18**	If object is a gallery or a story this variable holds its ID.
<%Ref%> **From version 7.2.0sr18**	If object is a gallery or a story this variable holds its Ref.

NewsItemExternalLinkItem.pbo

Used with the field

```
<%NewsItemExternalLinkItem%>
```

in NewsItemExternalLinks.pbo.

<%Counter%>	This field starts with the value 1 and is increased by one for each link.
<%ImageLink%>	Contains the link to the photo defined in Story External Links in the Edit interface.
<%LinkDescription%>	The description of the link.
<%LinkID%>	The ID of the link.
<%LinkMainCategory%>	The main category for the link.
<%LinkRegion%>	The region for the link.
<%LinkSubCategory%>	The subcategory for the link.
<%LinkSubRegion%>	The subregion for the link.
<%LinkTitle%>	The title of the link.
<%LinkURl%>	The link's URL.
<%LogoLink%>	Contains the link to the logo defined in Story External Links in the Edit interface.
<%TotalCount%>	Contains the total number of items returned.

NewsItemExternalLinks.pbo

Used to show links to external objects that are linked to the story.

<%NewsItemExternalLinkItem%>	Shows external links.

NewsItemExtraPictureItem.pbo

<%AltText%>	Outputs the information from the image Alt text field.
<%ArtNo%>	Contains the image ID number.
<%Caption%>	The caption for the picture.
<%CaptionPreformated%>	The title for the picture with font bold formatted.
<%CaptionTitle%>	The title for the picture.
<%Category%>	The image category.
<%Counter%>	Contains the picture counter. It starts on 1 for the first picture.
<%DesignClass%>	Contains the design class used.
<%Height%>	Contains the height of the picture.
<%Img%>	HTML image tag to make sure the image is displayed.
<%ImgKeyParams%>	Contains the key parameters for the picture. These parameters are used when you want to use functions like zoom. It is usually used like this: `<a href="/apps/pbcs.dll/misc?url=/templates/zoom.pbs&<%ImgKeyParams%>">Zoom</a>` This function is described in the section pbs link and images.
<%ImgLink%>	Contains the URL that is used as source for the **<img>** tag. It is possible to use the options **Title=1** and **Logo=/somefile.gif** in conjunction with this field so that the tag can be used like this: `<img src="<%ImgLink%><%MaxWidth%>&title=1&logo=/graphics/logo.gif">`
<%IsLastItem%>	This field has the value 1 on the last picture.
<%ISODate%>	The date of the image.

<%Orientation%>	This field contains the value
	`L`
	if the image orientation is landscape and
	`P`
	if it is portrait.

<%Photo%>	The photo byline.
<%Site%>	The two letter site code.
<%Width%>	Contains the width of the picture.
<%StorySite%> **From version 6.6.0sr43, 6.8.0sr8**	Story site code
<%StoryId%> **From version 6.6.0sr43, 6.8.0sr8**	Story ID
<%StoryCategory%> **From version 6.6.0sr43, 6.8.0sr8**	Story category
<%StoryDate%> **From version 6.6.0sr43, 6.8.0sr8**	Story date
<%TotalCount%> **From version 6.9.0sr1**	Contains the total number of extra pictures.
<%ImageLink%> **From version 6.6.0sr43, 6.8.0sr8**	Same as ImgLink

NewsItemExtraPictures.pbo

<%Count%>	Shows the number of pictures.
<%DesignClass%>	Contains the design class used.
<%Items%>	Outputs the items.

NewsItemForumConnectionItem.pbo

Used with the field <%ForumConnectionItems%> in NewsItemInternalLinks.pbo.

The file uses the same fields as NewsItemConnectionItem.pbo except <%IS....%> fields are not available.

<%ObjectClass%>	If you need to use the current object class with the forum template, add it to the link like this: `<%Link%>&ObjectClass=<%ObjectClass%>`

NewsItemForumGroupLink.pbo

Used to link directly to a story forum from the Newslist.

<%Link%>	The link to the forum.

NewsItemGalleryConnectionItem.pbo

Used with the field <%GalleryConnectionItems%> in NewsItemInternalLinks.pbo.

The file uses the same fields as NewsItemConnectionItem.pbo except <%IS....%> fields are not available.

NewsItemGeoData.pbo

The following list of parameters is available for use in the object file.

<%Items%>	This tag outputs all the items
<%Counter%>	number of Geo data selected

NewsItemGeoDataItem.pbo

The following object variables are available in this object file

<%Name%>	Name of the location
<%Location%>	The address of the location
<%Latitude%>	The latitude of the location
<%Longitude%>	The longitude of the location
<%Counter%>	item number

NewsItemInternalLinks.pbo

Used to show links to other internal objects that are linked to the story.

<%ArticleConnectionitems%>	Shows links to connected stories.
<%Connectionitems%>	Shows internal links to both stories and photo galleries.
<%ForumConnectionitems%>	Shows links to connected forums.
<%GalleryConnectionitems%>	Shows links to connected galleries.

NewsItemKeywordLinkItem.pbo

Please see KeywordLinkItem.pbo for information about available fields.

NewsItemKeywordLinks.pbo

Please see KeywordLinks.pbo for information about available fields.

NewsItemMultiMediaExtension.pbo

NewsItemMultiMedia[Extension].pbo

Used to link to directly connected multimedia objects. [Extension] is replaced by the media type extension.

<%Description%>	The description of the media object (listed as 'title' in the edit interface).
<%FileName%>	The file name.
<%FileSize%>	The file size.
<%Link%>	The link for direct download of the object.
<%PopupLink%>	The link for the pop-up multimedia window.
<%StreamLink%>	The link for streaming the object.
<%MediaTypeExt%> **From version 6.6.0sr43, 6.8.0sr8**	The extension of media file

NewsItemParagraphItem.pbo

The following list of parameters is available for use in the object file.

<%Heading%>	The Sub-Heading of paragraph.
<%Body%>	The body of paragraph.
<%Counter%>	Paragraph number
<%ObjectClass%>	Objectclass defined on the <pbs:newslist> tag
<%CaptionHead%>	Image Caption
<%CaptionBody%>	Image Text
<%CaptionByline%>	Image Byline
<%ImageWidth%>	Image Width
<%ImageHeight%>	Image Height
<%ImageOrientation%>	Image Orientation (P or L)
<%ImageLink%>	Image URL
<%AltText%>	Outputs the information from the image Alt text field.
<%SeoFilename%	The SEO friendly file name specified for the image if any is set.
<%StorySite%> **From version 6.6.0sr43, 6.8.0sr8**	Story site code
<%StoryId%> **From version 6.6.0sr43, 6.8.0sr8**	Story ID
<%StoryCategory%> **From version 6.6.0sr43, 6.8.0sr8**	Story category
<%StoryDate%> **From version 6.6.0sr43, 6.8.0sr8**	Story date

NewsItemTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

NewsItemTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

NewsItemTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

NewsItemTaxonomyWordItem.pbo

The following list of parameters is available for use in the object file.

<%id%>	Contains the ID of the taxonomy word.
<%name%>	Contains the name of the taxonomy word.

NewsItemTaxonomyWords.pbo

This file contains the items of the taxonomy words.

<%Items%>	This tag outputs all the items.

NewsItemThemeLinkItem.pbo

Please see ThemeLinkItem.pbo for information about available fields.

NewsItemThemeLinks.pbo

Please see ThemeLinks.pbo for information about available fields.

RelStoriesItem.pbo

Used with **RelTaxonomy** when showing related stories based on taxonomy.

<%ISODate%>	The publication ISOdate for the story.
<%Date%>	The publication date for the story.
<%Heading%>	The heading of the linked article.
<%Leadin%>	Outputs the summary if the leadin is larger than 0.
<%Link%>	The address of the connected article.
<%SEOLabel%> **From version 7.0.0sr11**	The SEO label of the connected article.

RelStoriesItems.pbo

Used with **RelTaxonomy** when showing related stories based on taxonomy.

<%ItemCount%>	The number of stories in the list.
<%Stories%>	The list of stories.

URL Parameters and overview tags

You can control what the page displays by putting parameters in the URL. This is done by adding **&params=** to the end of the URL.

Calling a Newslist with the URL parameters:

```
&Params=StDato=20050730,SlDato=20060731
```

sets a start date and an end date for the Newslist stories. If nothing is specified, a default range will be used. The default maximum range is 30 days (StDato - SlDato). This can be changed in the ini file. Please see MaxDaysForNewslistDateRange.

StDato	Used to specify the start date. Use YYYYMMDD format.
SlDato	Used to specify the end date. Use YYYYMMDD format.
Slstdato	An alternative way to set the start and end date of the story selection. Possible values could be **SISTE12T**, **SISTE24T**, or **SISTE14D**. If the value given is not one of these three, then it is assumed that the given value is a date in YYYYMMDD format and it will select stories from only that day.

Another option in the URL is to set the profile. When you place a profile in the **Newslist** tag, the profile selected on the **Category** settings page in the edit interface will be used. You can override this profile by entering

```
&Profile=XXX
```

in the URL, where XXX is the profile number. You can specify the profile directly by entering

```
Profile=XXX
```

in the tag. This will lock the tag to display stories matching profile number XXX. You can limit the selection of stories using categories with profiles. This means that stories matching a profile also will have to match the category.

pbs related

<pbs:related>

The theme overview is used for showing all stories with a common theme. Usually, only 5 to 10 links to stories with the same theme are shown when you view the story. The theme overview enables you to show all the stories with the same theme.

This function uses the template **gen_tovr.pbs**. The link to this page is provided by `<%MoreLink%>` in the listing of related stories on the story.

Options

Antall	The number of stories to be shown. By default, the value is all stories will be shown. Enter antall=x where x is any number between 1 and 9999.
Bgcolor	Sets the background color. This color is set in the standard HTML way. This option is ignored when object files are used.
Color	Sets the text color when object files are not used. This color is set in the standard HTML way.
Daterange	By default, only stories from the last 180 days are shown. Enter months=x to show stories from the last x months.

Leadin	This option determines how many characters of the lead-in will be shown. By default, the entire lead-in is shown. Enter
	`leadin=x`
	where x is any value from 0 to 9999. Setting
	`leadin=0`
	turns off the lead-in.

Objectclass	Sets the object class for the tag. By default, the tag will use the ThemeLinks.pbo and ThemeLinkItem.pbo object files. Enter `objectclass=x` where x is any number from 1 to 99 to change the object class.

ThemeLinkItem.pbo and overview tags

<%\Date%>	The story's date. Formatting is controlled by the **Dateformat** option.
<%IsoDate%>	The date in ISO format.

<%Leadin%>	The summary of the story. The amount of text shown is controlled by the **Leadin** option.
<%Link%>	The link to the story.
<%\Title%>	The title of the story.

An example of using these options in the object file might be the following:

```
<li type=square> <a href="<%Link%>"><%\Title%></a> <%\Date%>
```

ThemeLinks.pbo and pbs related

<%Items%>	This field outputs all the items.
<%Theme%>	The name of the theme.

pbs subcategories

<pbs:subcategories>

This tag is used to list subcategories. It can also list the category's stories.

The tag with options can look like this:

```
<table>
<pbs:subcategories force main columns=3 description=0 class=2>
</table>
```

This tag must be placed inside a table if you use multiple columns.

Options

Note that the system will use object files if they exist. All options except class are also available when you are not using object files.

Category	Used to list a specific subcategory.
Class	Determines which object class the tag should use. By default, SubcatSimpleLink.pbo is used. By setting this option to 1, the system will use SubcatSimpleLink1.pbo instead.
Columns	This option controls how many columns the list will have.
Count	Determines the number of stories shown. 0 (zero) outputs all stories.
Force	Makes the system list all subcategories and ignore the **Active** flag in the category register.
Main	Makes the system list all subcategories of the main category being viewed. If this option is not set, only subcategories of the category being viewed will be shown.
Maincategory	Used to list subcategories of the selected main category. The option is set like this: `Maincategory=NEWS`

Showart	Retrieves the stories from the subcategory. By default, this option is off. Change this option setting to On to retrieve stories from the subcategory.
Visref	Determines how many characters of the lead-in will be shown. The option is set like this: `Visref=200`

SubcatSimpleLink.pbo

The object file is output once per category.

<%Article%>	Lists the stories for the subcategory.
<%CategoryName%>	The name of the category.
<%Link%>	The link to the sectionfront for the category.

Images

This chapter contains general information about images in the system and information about the permanent gallery. Please note that SAXOTECH Online has a permanent gallery and a story based gallery. The two galleries are accessed at different locations in the editing interface.

- Access control for images
  - PhotoGalleryExtLogin.pbo
- Story-linked photo gallery
- Generate graphical text
- Order picture
- pbs link and images
- pbs photogallery
  - ArticleConnectionItem.pbo and images
  - ArticleConnections.pbo and images
  - PhotoGallery.pbo
  - PhotoGalleryAccessControls.pbo
  - PhotoGalleryAccessControlsItem.pbo
  - PhotoGalleryItemExtraFieldItem.pbo
  - PhotoGalleryItemExtraFields.pbo
  - PhotoGalleryItemTaxonomy.pbo
  - PhotoGalleryItemTaxonomyItem.pbo
  - PhotoGalleryTaxonomy.pbo
  - PhotoGalleryTaxonomyItem.pbo
  - PhotoGalleryTaxonomyItemPathItem.pbo
- pbs photogallery-latest
  - PhotoGalleryLatest.pbo
  - PhotoGalleryLatestItem.pbo
  - PhotoGalleryLatestItemAccessControls.pbo
  - PhotoGalleryLatestItemAccessControlsItem.pbo
  - PhotoGalleryLatestItemExtraFieldItem.pbo
  - PhotoGalleryLatestItemExtraFields.pbo
  - PhotoGalleryLatestItemListTaxonomy.pbo
  - PhotoGalleryLatestItemListTaxonomyItem.pbo
  - PhotoGalleryLatestItemListTaxonomyItemPathItem.pbo
  - PhotoGalleryLatestItemTaxonomy.pbo
  - PhotoGalleryLatestItemTaxonomyItem.pbo
- pbs photogallery-thumbnails
  - PhotoGalleryThumbnailItem.pbo
  - PhotoGalleryThumbnails.pbo
  - PhotoGalleryThumbnailsAccessControls.pbo
  - PhotoGalleryThumbnailsAccessControlsItem.pbo
- pbs ptshowimage
  - ShowImage.pbo
- Photo album export
  - ExpPhotoAlbum.pbo
    - ExpPhotoAlbumAccessControlsItem.pbo
  - ExpPhotoAlbumItem.pbo
  - ExpPhotoAlbums.pbo
- Set picture as desktop background
- Watermark images
- pbs ptimages
  - ImageItem.pbo
  - ImageItemRowEnd.pbo
  - ImageItemRowStart.pbo

Access control for images

It is possible to control the access to images. In order to give or limit access to certain galleries, you can use Access Control. For more information on Access Control, please see the *SAXOTECH Online User Manual.*

To set up Access Control in the ini file, please see EXTLOGIN section#AccessControl.

Template

Access Control uses the template **gen_gallery_extlogin.pbs**.

PhotoGalleryExtLogin.pbo

<%StandardText%>	The text to be output. For example it could say "You do not have access to this gallery, but you can purchase access, by clicking..."

Story-linked photo gallery

The photo gallery connected to a story is usually used when you have too many photos for the story. For example, you have a travel story from an exotic location and you would like to show 20 photos. How to use the story-linked photo gallery is described in the *SAXOTECH Online User Manual.*

Linking to a photo gallery is documented in the section pbs photogallery-links.

Generate graphical text

The **Pbcsi.dll**, also known as the Image dll, has the ability to generate GIF images containing a text string. This will allow you to build graphical headers for your pages without having to edit the images directly.

Tag

This function has no tag. You pass parameters to the **image dll**. Example:

```
<img src="/apps/pbcsi.dll/text?
text=Test&Fontname=Arial&Fontsize=16&Width=300&Height=-1
&FontColor=FFFFFF&BGColor=000000&XOffset=8&YOffset=0">
```

Template

This function can be used on all pages.

Options

BGColor	Specifies the background color. The color is set using an RGB triplet.
Bold	Makes the text bold. `Enter bold=1` to enable this function.
Fontcolor	Specifies the font color. The color is set using an RGB triplet.
Fontname	Specifies the font. Only fonts installed on the image server can be used.
Fontsize	Specifies the font size.
Height	Specifies the height of the image. Setting this value to `-1` will make the system scale the image.
Italic	Makes the text italic. Enter `italic=1` to enable this function.
Text	Sets the text string.

Transparent	Makes the background transparent. Enter `transparent=1` to enable this function.
Underline	Makes the text underlined. Enter `underline=1` to enable this function.
Width	Specifies the width of the image. Setting this value to `-1` will make the system scale the image.
Xoffset	Specifies the padding in pixels on the left hand side of the text string.
Yoffset	Specifies the padding in pixels on top of the text string.

Example code

This document will allow you to experiment with the image functionality:

```
<HTML>
<BODY BGCOLOR=#EEEEEE>
<SCRIPT Language="Javascript">
<!--
function Update() {
var frm=document.Settings;
var img=document.getElementById("image");
var
url='/apps/pbcsi.dll/text?text='+escape(frm.Text.value)+'&Fontname='+escape(frm.Font.v
alue)+'&Fontsize='+frm.Fontsize.value+'&Width='+frm.Width.value+'&Height='+frm.Height.
value+'&FontColor='+frm.Fontcolor.value+'&BGColor='+frm.BGColor.value+'&XOffset='+frm.
XOffset.value+'&YOffset='+frm.YOffset.value;
if (frm.Bold.checked) {url = url + '&Bold=1';}
if (frm.Italic.checked) {url = url + '&Italic=1';}
if (frm.Underline.checked) {url = url + '&Underline=1';}
if (frm.Transparent.checked) {url = url + '&Transparent=1';}
img.src = url;
}
-->
</SCRIPT>
<FORM NAME="Settings">
<TABLE Border=0>
<TR>
<TD>Width</TD><TD><INPUT Type="Text" Name="Width" Size=5 value="300"></TD>
</TR>
<TR>
```

```
<TD>Height</TD><TD><INPUT Type="Text" Name="Height" Size=5 value="-1"></TD>
</TR>
<TR>
<TD>Text</TD><TD><INPUT Type="Text" Name="Text" Size=40 value="Test"></TD>
</TR>
<TR>
<TD>Fontname</TD><TD><INPUT Type="Text" Name="Font" Size=40 value="Arial"></TD>
</TR>
<TR>
<TD>Fontsize</TD><TD><SELECT Name="Fontsize"><OPTION Value="-1">Automatic<OPTION
Value="10">10pt<OPTION Value="12">12pt<OPTION Value="16">16pt<OPTION
Value="20">20pt<OPTION Value="24">24pt<OPTION Value="30">30pt<OPTION
Value="36">36pt</SELECT></TD>
</TR>
<TR>
<TD>Fontcolor</TD><TD><INPUT Type="Text" Name="Fontcolor" Size=8 value="FFFFFF"></TD>
</TR>
<TR>
<TD>Backgroundcolor</TD><TD><INPUT Type="Text" Name="BGColor" Size=8
value="000000"></TD>
</TR>
<TR>
<TD>Text X-Offset</TD><TD><INPUT Type="Text" Name="XOffset" Size=5 value="0"></TD>
</TR>
<TR>
<TD>Text Y-Offset</TD><TD><INPUT Type="Text" Name="YOffset" Size=5 value="0"></TD>
</TR>
<TR>
<TD COLSPAN=2>Bold<INPUT TYPE=Checkbox Name=Bold> Italic<INPUT TYPE=Checkbox
Name=Italic> Underline<INPUT TYPE=Checkbox Name=Underline> Transparent<INPUT
TYPE=Checkbox Name=Transparent> </TD>
</TR>
<TR>
<TD COLSPAN=2><hr></TD>
</TR>
<TR>
<TD COLSPAN=2 ALIGN=Right><INPUT Type="Button" Value="Update" onClick="Update()"></TD>
</TR>
</TABLE>
</FORM>
<img id=image src="/g/t.gif">
```

```
</BODY>
</HTML>
```

Order picture

If you would like to have an order form in which web guests can purchase prints of pictures, include the following link in your ArticlePicture.pbo:

```
<a href="<%pbcs.dll%>/misc?url=/misc/orderpicture.pbs&<%ImgKeyParams%>">
Order Picture
</a>
```

Template

In this example, the file **orderpicture.pbs** in the /misc folder is used.

Form

The form is setup with the following line:

```
<FORM METHOD="POST" ACTION="/apps/xsendmail.dll">
```

Hidden fields

These fields are used to set options.

Article URL	Sets the URL to the story that is linked to the photo ordered by the web guest. You must replace www.yoursite.com with the address to your website. `<INPUT TYPE=hidden NAME="Article URL" VALUE="<a href='http://www.yoursite.com<pbs:!link artikkel inherit href>'>Click here</a>">`
From	Sets the from address for the email. `<INPUT TYPE=hidden NAME="FROM" VALUE="Webserver <webserver@publicus.com>">`

Lang	Sets the language of system messages. `<INPUT TYPE=hidden NAME="LANG" VALUE="ENG">`
Photo URL	Sets the URL to the photo the web guest has ordered. You must replace www.yoursite.com with the address to your website. `<INPUT TYPE=hidden NAME="Photo URL" VALUE="<a href='http://www.yoursite.com<pbs:!link inherit href text=img: ondate=0>'>Click here</a>">`
Redirect URL	Sets the URL to which the web guest will be sent after the form is filled out correctly. `<INPUT TYPE=hidden NAME="REDIRECT_URL" VALUE="/apps/pbcs.dll/misc?url=/misc/orderpicture_ok.pbs">`
Required	Determines which fields in the form are required. `<INPUT TYPE=hidden NAME="REQUIRED" VALUE="Name,Address,CityStateZip,Email">`
Subject	Sets the subject of the email sent to your staff. `<INPUT TYPE=hidden NAME="SUBJECT" VALUE="Order a picture.">`
Title	Sets the title of the table that contains the information in the email. `<INPUT TYPE=hidden NAME="TITLE" VALUE="Order a picture">`

To	Sets the recipient of the e-mail. This is the e-mail address of the person handling picture orders.`<INPUT TYPE=hidden NAME="TO" VALUE="eh@saxotech.no">`

Input fields

Include the required input fields, like name, address and phone number. The fields are defined in the following way:

```
<input type="text" name="Name" size="20" maxlength="40">
```

Replace

```
"Name"
```

with the name of the field.

Buttons

You must have a **Submit** button in the form.

Thank you page

This template must be called **orderpicture_ok.pbs** and be put in the /**misc** folder if you are using the paths in this example. This page can contain whatever you like.

pbs link and images

<pbs:link>

This function is used to give the web guest the ability to zoom in on a picture in a story. This function uses the `<pbs:link>` tag. You must use the tag like this:

```
<img src="<pbs:!link inherit ondate=0 href text=img:>&MaxW=580&title=1">
```

Template
The template used to zoom can be called whatever you like. You must specify the desired template in the link.

Link to zoomed picture
To link to the template used for the zoomed picture, you must place the following link in the object file with picture:

```
<a href="/apps/pbcs.dll/misc?url=/
```

```
*misc/zoom.pbs*
```

```
&<%ImgKeyParams%>">Zoom</a>
```

It is usually used in a pop-up window. If you would like to use this function in a pop-up window, you must use the following JavaScript:

```
<a
href="javascript:NewWindow(460,300,'/apps/pbcs.dll/misc?url=/misc/zoom.pbs&<%ImgKeyPar
ams%>');">Zoom</a>
```

Options

Logo	Please see images_Watermark_Images#Image_Watermark for more information.
Title	Please see images_Watermark_Images#Text_Watermark for more information.

Width	You can set the width of the picture by appending
	`&Maxw=X`
	where
	`X`
	is the width of the image, in pixels.

pbs photogallery

<pbs:photogallery>

Options

ObjectClass	Sets the object class.
ShowExtraFields	Use to show extra fields - default is 0. `showextrafields=1`
ShowTaxonomy	Use to show taxonomy words - default is 0. `showtaxonomy=1`
IgnoreStatus	Use if publish status is to be ignored - default is 0. `ignorestatus=1` Using it will display published and unpublished galleries.

The template for the photo gallery is called **gen_gallery.pbs**. It must be placed in the template folder. It is possible to create templates for each category. Link to the gallery page from a story, or by using the pbs photogallery-latest tag.

If you want to link to the gallery manually, you must specify all parameters like this:

```
<a
href="apps/pbcs.dll/gallery?Date=20030609&Category=REPOSITORY&lopenr=609001&Ref=PH">Li
nk</a>
```

This tag can also be used to display a zoomed version of the photo in the gallery. This function uses the template **gen_galleryzoom.pbs**. This function uses the same options and object files below.

ArticleConnectionItem.pbo and images

This file is output for each story that is linked to the photo gallery. This object file, along with ArticleConnections.pbo, is used when <%ArticleConnectionItems%> is placed in the object file PhotoGallery.pbo.

<%Heading%>	The title of the story.
<%Link%>	The link to the story.

ArticleConnections.pbo and images

Used if links to the connected stories should be output. This object file is used when <%ArticleConnectionItems%> is placed in the object file PhotoGallery.pbo.

<%ConnectionItems%>	Outputs each link.

PhotoGallery.pbo

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of PhotoGallery AccessControls.pbo
<%ArticleConnectionItems%>	Outputs links to all stories that are connected to the photo gallery.
<%Byline%>	The byline for the photo.
<%Caption%>	The caption for the photo.
<%CurrentPicture%>	The number of the picture being viewed.
<%FirstURL%>	A link to the first photo in the album, except when viewing the first photo. It is usually used like this: `[%{FirstURL} <a href="<%FirstURL%>">First</a> %|% First %]`
<%Field_FieldName%>	The fields are available in the following format <%Field_TheNameOfTheField%>
<%GalleryText%>	The gallery description.
<%GalleryTitle%>	The title for the gallery.
<%GalleryZoomLink%>	A link to the same photo on a different template. The field is described in pbs link.
<%Height%>	The height of the picture.
<%ImgParams%>	This field contains all parameters for the image. The field can be used to link to an order form for the image.

<%JumpMenuOptions%>	Builds a drop-down menu, which enables web guests to jump to any picture in the gallery. It is built like this:
	
	`<form>`
	`Jump to:`
	`<select`
	`onChange="`
	`document.location.href=this.options[this.selectedIndex].value;`
	`this.options[this.selectedIndex].text='`
	`...Wait...';">`
	`<%JumpMenuOptions%>`
	`</select>`
	`</form>`
	
	When you use this option, it is important that the `<select>` tag is written on one line

<%LastURL%>	A link to the last photo in the album, except viewing the last photo. It is usually used like this `[%{LastURL} <a href="<%LastURL%>">Last</a> %|% Last %]`
<%NextURL%>	A link to the next photo in the album, except when viewing the last photo. It is usually used like this: `[%{NextURL} <a href="<%NextURL%>">Next</a> %|% Next %]`
<%PictureCount%>	The total number of photos in the gallery.
<%PictureURL%>	The URL for the picture. This field is usually used in the following piece of code: `<img src="<%PictureURL%>&Maxw=400&Maxh=400">`
<%PreviousURL%>	A link to the previous photo in the album, except when viewing the first photo. It is usually used like this: `[%{PreviousURL} <a href="<%PreviousURL%>">Previous</a> %|% Previous %]`
<%Title%>	The title of the picture.

<%Width%>	The width of the picture.
<%UserSubmitDeleteLink%>	URL that if visited will delete the story - in case the story is a user submitted story and the user is the creator.
<%GalleryID%>	Contains the lopenr / StoryID of the gallery.
<%GalleryCategory%>	Same as Category.
<%GalleryDate%>	Contains the date of the gallery.
<%GUID%> **From version 6.7.0**	The gallery's GUID.
<%Location%>	Contains the location field value.
<%PhotoGalleryStatus%>	Contains the publish status of the photo gallery. It can be published (=1) or unpublished (=0). This is only returned if the photogallery-latest tag has the ignorestatus=1 parameter.
<%CreatedByUserID%>	Contains the creator's user ID if photo gallery is a blog entry.
<%CreatedByUserFirstName%>	Contains the creator's first name if photo gallery is a blog entry.
<%CreatedByUserSurName%>	Contains the creator's last name if photo gallery is a blog entry.
<%CreatedByUserEmailAdr%>	Contains the creator's email address if photo gallery is a blog entry.
<%CreatedByUserNickName%>	Contains the creator's nickname if the photo gallery is a blog entry.
<%BlogEntry%>	Whether the photo gallery is user generated (=1) or not (=0).
<%ISOCreatedDate%>	ISO formatted created date.
<%ISOModifiedDate%>	ISO formatted modified date.
<%Taxonomy%>	Contains the taxonomy words of the photo gallery. See PhotoGalleryItemTaxonomy.pbo
<%TaxonomyList%> **From version 6.9.0sr10**	Contains the taxonomy words of the photo gallery. See PhotoGalleryTaxonomy.pbo
<%ExtraFields%>	Contains the extra fields of the photo gallery. see PhotoGalleryItemExtraFields.pbo
<%CategoryID%>	Contains the category ID of the photo gallery.
<%CategoryDescription%>	Contains the category name.
<%AvgRanking%>	Contains the average ranking of a Photo.
<%RankingLink%>	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL '<%RankingLink%>' + '&rating=' + y Where y is the ranking given by a user.
<%TotalNoofVotes%>	Contains the total number of votes on the Photo.

PhotoGalleryAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of PhotoGalleryAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

PhotoGalleryAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.

<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

PhotoGalleryItemExtraFieldItem.pbo

<%ExtraFieldName%>	Contains the name of the extra field.
<%ExtrafieldValue%>	Contains the value of the extra field.

PhotoGalleryItemExtraFields.pbo

<%Items%>	Contains the extra field items.

PhotoGalleryItemTaxonomy.pbo

<%Items%>	Contains the taxonomy words.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

PhotoGalleryItemTaxonomyItem.pbo

<%TaxonomyID%>	Contains the taxonomy ID.
<%TaxonomyWord%>	Contains the taxonomy word.

PhotoGalleryTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.

PhotoGalleryTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

PhotoGalleryTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

pbs photogallery-latest

This function output links to the newest photo galleries in the system. The tag can be used on all templates.

Options

Category	Used when you want to filter on categories. It is possible to use the pattern matching. For example,
	`category="NEWS,SPORTS"`
	selects galleries from both the *News* and *Sports* categories as in this example:
	`category="news%,sports0[1-3],arts[^4],opinion"`
	The options are:
	`category="news%,sports%"`
	Selects galleries from all News and Sports categories.
	`category="news0[5-6],sports0[1-2]"`
	Selects galleries from the categories news05, news06, sports01 and sports02. Please note that writing `category="CATEGORYID[05-06|05-06]"` is not supported.
	`category="news0[^5],sport0[^1]"`
	Selects galleries from all categories except for news05 and sports01.

Count	By default, the 5 latest galleries are shown. Enter `count=x` where x is between 1 and 99 to show the selected number of photo galleries.

CreatedBy	Used to filter by gallery creator. Enter
	`createdby=asdf`
	To display the galleries created by the journalist with username 'asdf'. Enter
	`createdby=#currentuser#`
	To display the user's own stories.

Days	By default, this option is not set. Enter `days=x` where x is between 1 and 99.
ExactTaxonomyMatch **From version 6.9.0sr15**	This settings controls whether the entire taxonomy tree is used when using **Taxonomywords** parameters. By default the entire taxonomy tree is used. This means that galleries which have the taxonomy word identified in the photogallery-latest including the parent structure will be selected. If this settings is set to 1 then only galleries which have specified taxonomy ids assigned to the directly will be selected.
Excltaxonomywords	Exclude a specific taxonomy word. `Excltaxonomywords="11,12"`

ExcludeCategories
From version 6.6.0sr9
	Used when you want to exclude galleries from the list based on categories. To exclude multiple categories, enter them into the tag. For example,
	
	`ExcludeCategories="NEWS,SPORTS"`
	
	excludes galleries from both the *News* and *Sports* categories as in this example:
	
	`ExcludeCategories="news%,sports0[1-3],arts[^4],opinion"`
	
	The options are:
	
	`ExcludeCategories="news%,sports%"`
	
	Excludes galleries from all News and Sports categories
	
	`ExcludeCategories="news0[5-6],sports0[1-2]"`
	
	Excludes galleries from the categories news05, news06, sports01 and sports02. As with the category option, the range used must be one character. Writing `ExcludeCategories="CATEGORYID05-06"` is not supported.

Fields	Determines which extra fields is searchable. All fields in the field definitions can be used. The option is set like this: `<pbs:photogallery-latest fields="Topic,Location">`

FIELDSearch	Searches a specified field. Replace FIELD with the name of the field. The option is used like this:
	
	`FIELDSearch="text"`
	
	Example:
	
	`<pbs:photogallery-latest fields="Topic,Location" TopicSearch="Football" LocationSearch="NY">`

IgnoreStatus	Use if the publish status should be ignored. The default is 0. `ignorestatus=1` Using it will display published and unpublished galleries.
Incltaxonomywords	Include a specific taxonomy word. `Incltaxonomywords="11,12"`
Kwnot **From version 6.9.0sr15**	This option separates the keywords (**Taxonomywords**) with NOT. The default option for keywords is to use the AND separator. Entering `kwnot="1"` in the tag determines that the keywords will be separated with NOT.
Kwor **From version 6.9.0sr15**	This option separates the keywords (**Taxonomywords**) with OR. The default option for keywords is to use the AND separator. Entering `kwor="1"` in the tag determines that the keywords will be separated with OR.
Objectclass	Sets the object class.

Order	Sets the sort order for the listing of photo galleries. By default the list is sorted by creation date. Enter
	
	`order=modified`
	
	to sort the list by the photo albums' modification date. Enter
	
	`order=published`
	
	to sort the list by the photo albums' publication date.
	
	From version 7.2.0sr23 it is also possible to order by priority, plainpriority and profilepriority.
	profilepriority requires the filter parameter, **Profile**.
	priority is a mixed sort of the two other priority sorts, where the priority for each item is its profilepriority if available, otherwise its plainpriority.

Profile
From version 7.2.0sr23
	You can filter items a specific profile. Examples:
	`<pbs:photogallery-latest profile="1142">`
	`<pbs:photogallery-latest profile="#profile#">`
	As can be seen the #profile# is available, specifying the context profile, e.g., found in the URL.

ShowEmpty	Use to show empty galleries in the list (=1). The default is 0 (dont show).
ShowExtraFields	Use to show extra fields. The default is 0. `showextrafields=1`
ShowTaxonomy	Use to show taxonomy words. The default is 0. `showtaxonomy=1`

Taxonomywords
From version 6.9.0sr15
	You can output galleries matching specific taxonomy words. By default, this option is not used. Example:
	`<pbs:photogallery-latest Taxonomywords="12,14" />`
	When you specify multiple words, they will be separated by AND. By adding Kwor=1 to the tag, the terms will be separated by OR. Example:
	`<pbs:photogallery-latest taxonomywords="12,14" kwor="1" />`
	Please also refer to the **ExactTaxonomyMatch** parameter.
	The Taxonomywords parameter should not be used with the **Excltaxonomywords** or **Incltaxonomywords** parameters.

PhotoGalleryLatest.pbo

This file is output when the system shows the latest galleries.

<%GalleryItems%>	This field outputs all items.

PhotoGalleryLatestItem.pbo

Item	Description
<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of PhotoGalleryLatestItemAccessControls.pbo
<%Category%>	The category for the photo gallery.

<%Counter%>	A counter for the gallery.
<%CreatedLongDate%>	The date on which the gallery was created, in long format.
<%CreatedShortDate%>	The date on which the gallery was created, in short format.
<%ImgLink%>	The img src for the first photo in the gallery.
<%Link%>	The link to the photo gallery.
<%ModifiedLongDate%>	The date on which the gallery was last modified, in long format.
<%ModifiedShortDate%>	The date on which the gallery was last modified, in short format.
<%PictureCount%>	The number of photos in the gallery.
<%TotalCount%> **From version 6.9.0sr2**	The total number of galleries returned.
<%Text%>	The description of the gallery.
<%ThumbnailLink%>	The link to the thumbnail version of the gallery.
<%Title%>	The title of the photo gallery.
<%UserSubmitDeleteLink%>	URL that if visited will delete the story - in case the story is a user submitted story and the user is the creator.
<%Field_FieldName%>	The fields are available in the following format <%Field_[TheNameOfTheField]%>
<%GalleryID%>	Contains the lopenr / StoryID of the gallery.
<%GalleryCategory%>	This corresponds to the Category ID in the edit interface: Admin > Categories > Category Form > Category ID field. This is **NOT** the same as <%Category%>. This value would look something like BIZ01 where as <%Category%> would be the human readable form, like "Business Local". Use <%GalleryCategory%> in URLs and in calls to other tags where there is a category parameter.
<%GalleryDate%>	Contains the date of the gallery.
<%GUID%>	The gallery's GUID.
<%Location%>	Contains the location field value.
<%PhotoGalleryStatus%>	Contains the publish status of the photo gallery. It can be published (=1) or unpublished (=0). This is only returned if the photogallery-latest tag has the ignorestatus=1 parameter.
<%CreatedByUserID%>	Contains the creator's user ID if photo gallery is a blog entry.
<%CreatedByUserFirstName%>	Contains the creator's first name if photo gallery is a blog entry.
<%CreatedByUserSurName%>	Contains the creator's last name if photo gallery is a blog entry.
<%CreatedByUserEmailAdr%>	Contains the creator's email address if photo gallery is a blog entry.
<%CreatedByUserNickName%>	Contains the creator's nickname if the photo gallery is a blog entry.
<%BlogEntry%>	Whether the photo gallery is user generated (=1) or not (=0).
<%ISOCreatedDate%>	ISO formatted created date.
<%ISOModifiedDate%>	ISO formatted modified date.
<%Taxonomy%>	Contains the taxonomy words of the photo gallery. See PhotoGallery LatestItemTaxonomy.pbo
<%TaxonomyList%> **From version 6.9.0sr10**	Contains the taxonomy words of the photo gallery. See PhotoGallery LatestItemListTaxonomy.pbo
<%ExtraFields%>	Contains the extra fields of the photo gallery. see PhotoGalleryLatest ItemExtraFields.pbo

<%AvgRanking%>	Contains the average ranking of a photo.
<%BayesianRanking%>	Bayesian ranking of a photo.
<%RankingLink%>	The RankingLink does not contain the actual ranking value, which must be added to the link using client side scripting. The following must be added to the link before submitting the link to the DLL: `'<%RankingLink%>' + '&rating=' + y` Where y is the ranking given by a user.
<%TotalNoofVotes%>	Contains the total number of votes on the photo.
<%CurrentFlowStatus%> **From version 6.9.0sr24**	Contains the current flow status of the gallery.
<%SingleSaleIdent%> **From version 6.9.0sr24**	Contains the single sale Status ID associated to a gallery.
<%Flow%> **From version 6.9.0sr24**	Contains the flow ID that is associated with a story in the edit interface. If there are no flows associated with the story, then this variable will be empty. This variable is used in conjunction with Access Control.

PhotoGalleryLatestItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of PhotoGallery LatestItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

PhotoGalleryLatestItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

PhotoGalleryLatestItemExtraFieldItem.pbo

In addition, field specific object files are available using PhotoGalleryLatestItemExtraFieldItem<FieldName>.pbo

<%ExtraFieldName%>	Contains the name of the extra field.
<%ExtrafieldValue%>	Contains the value of the extra field.

PhotoGalleryLatestItemExtraFields.pbo

<%Items%>	Contains the extra field items.

PhotoGalleryLatestItemListTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

PhotoGalleryLatestItemListTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

PhotoGalleryLatestItemListTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

PhotoGalleryLatestItemTaxonomy.pbo

<%Items%>	Contains the taxonomy words.

PhotoGalleryLatestItemTaxonomyItem.pbo

<%TaxonomyID%>	Contains the taxonomy ID.
<%TaxonomyWord%>	Contains the taxonomy word.

pbs photogallery-thumbnails

<pbs:photogallery-thumbnails>

This function uses the template **gen_tngallery.pbs**. This template is displayed by the action **/apps/pbcs.dll/tngallery**. Link to the gallery page from a story, or by using pbs photogallery-latest. If you want to link to the gallery manually, you must specify all parameters like this:

```
<a href="apps/pbcs.dll/tngallery?
*Date=20030609&Category=REPOSITORY&artno=609001&Ref=PH*
">
Link
</a>
```

Notice if neither rows or columns are set, the system will only show 9 pictures. In this case, set rows to 0, by doing that all pictures would be shown.

Options

Columns	Determines the number of columns in the gallery table. Enter `columns=x` where x is the number of columns in the gallery table. The default number of columns is set to 3 so when there is an image count for the gallery other than multiples of 3 there will be displayed extra blank images.
ID	Specifies a gallery when you are viewing the gallery on another template, or through another action. This option has no default. Set this option as follows: `id="<sitecode>,<date>,<category>,<number>"`
Objectclass	Sets the object class used by the tag.

Rows	Determines the number of rows in the gallery table. By default, this is set to 3 rows. Enter
	`rows=x`
	where x is the number of rows. Setting
	`rows=0`
	will display all photos on the same page.

PhotoGalleryThumbnailItem.pbo

Output for each photo in the gallery.

<%Byline%>	The photo credit.
<%Caption%>	The caption for the photo.
<%Counter%>	The counter for the photo.
<%GalleryID%>	The ID of the gallery (Lopenr).

<%GalleryItemLink%>	The link to the current photo in the browsable version of the gallery.
<%GalleryLink%>	The link to the first photo in the browsable version of the gallery.
<%GalleryZoomLink%>	A link to the current photo on a different template. The function is described in pbs link.
<%Height%>	The height of the photo.
<%ImgLink%>	The image src for the photo. It is used like this: `<img src="<%ImgLink%>&Maxw=200&Maxh=200">`
<%ISOModifiedDate%> **From versions 6.9.0sr34, 7.0.0sr10**	Last modified time in ISO datetime format (yyyymmddhhnnss) of the gallery image.
<%ItemNr%>	The image number within the gallery.
<%RowEnd%>	This field has the value 1 on the last photo on the row. The field is used to end the table row. `[%{RowEnd}</tr>%]`
<%RowStart%>	This field has the value 1 on the first photo on the row. The field is used to start the table row. `[%{RowStart}<tr>%]`
<%TableEnd%>	This field has the value 1 on the last photo. The field is used to end the table. `[%{TableEnd}</table>%]`
<%TableStart%>	This field has the value 1 on the first photo. The field is used to start the table. `[%{TableStart}<table>%{code]`
<%ThumbnailGallerylink%>	A link to the thumbnail gallery. This field is used to display photos from the gallery on other templates, such as directly in a story.
<%Title%>	The title of the photo.
<%Width%>	The width of the photo.

PhotoGalleryThumbnails.pbo

This file builds the wrapper for the gallery.

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of PhotoGallery ThumbnailsAccessControls.pbo
<%GalleryLink%>	A link to the browsable gallery. This field is used to display photos from the gallery on other templates, such as directly in a story.
<%GalleryTitle%>	The title for the album.
<%GalleryText%>	The description of the gallery.

<%ISOCreatedDate%> **From version 7.3.0sr3**	The created time in format: yyyymmddhhnnss.
<%ISOModifiedDate%> **From version 7.3.0sr3**	The modified time in format: yyyymmddhhnnss.
<%ISOPublishedDate%> **From version 7.3.0sr3**	The published time in format: yyyymmddhhnnss.
<%Items%>	Outputs all the items.
<%NextLink%>	The link to the next page.
<%PageStart%>	The number for the first photo on the page.
<%PageStop%>	The number for the last photo on the page.
<%PrevLink%>	The link to the previous page.
<%ThumbnailGalleryLink%>	A link to the thumbnail gallery. This field is used to display photos from the gallery on other templates, such as directly in a story.
<%Total%>	The total number of photos in the gallery.

PhotoGalleryThumbnailsAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of PhotoGallery ThumbnailsAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

PhotoGalleryThumbnailsAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

pbs ptshowimage

<pbs:ptshowimage>

This function is used to display one of the pictures in the gallery.

The tag might appear as follows:

```
<pbs:ptshowimage maximagewidth=468 showtitle=1 quality=100>
```

Templates

This option uses the template specified in the link to the page.

Options

Maximageheight	Sets the maximum height for the picture.
Maximagewidth	Controls the width of the picture. It must be present. If it is missing, the system will load the full image from the database, which could destroy your layout.
Quality	Controls the quality of the picture. The higher the number, the better the quality. The default quality setting is 85. This option can be set from 1 to 100, where 1 is the lowest quality and 100 is the best quality.
Showtitle	Determines whether the name of the picture will be overlayed on the picture. By default, the title will not be output. Enter `showtitle=1` to show the title.

Link from gallery

The link from the gallery is described in the section ImageItem.pbo.

Link from story

To link from the story to the zoomed picture, you must put the following code in the ArticlePicture.pbo:

```
<a
href="<%pbcs.dll%>/misc?url=/misc/photos_single.pbs&IDato=<%ISODate%>&IKategori=<%Cate
gory%>&ID=<%ArtNo%>">Zoom</a>
```

The URL must match the URL you have made for the zoom template. This function can only be used on the main picture.

ShowImage.pbo

This file is used by pbs ptshowimage and it can contain the following variables:

<%Byline%>	The photo credit.
<%Caption%>	The caption for the image.
Desktop background	To allow web guests to use the picture as a desktop background, you must use this piece of code to link to the function: `<a href="javascript:NewWindow(460,300,'/apps/pbcs.dll/misc?URL=/misc/photos_desktop.pbs &NoMal=1&IDato=<%IDato%>&IKategori=<%IKategori%>&ID=<%ID%>');">Set as desktop background</a>` You can change the name of the template used (bold in the example above), but not any of the other parameters. This function is described in the section Set Picture as Desktop Background.

<%ImagePath%>	The path to the image. It is used like this: `<img src="<%ImagePath%>" alt="" border="0">`
<%\Title%>	The title of the image.

Photo album export

This functionality enables export of one or more photo albums from the system. Output is static image files and an XML file describing the photo album structure and meta data to be delivered to an FTP server.

The exported photo album can be imported to other sites using the Album Importer.

Publicus.ini settings

To activate the permanent gallery, you must have the following setting in the Publicus.ini file in the GENERAL section section:

```
[General]
PhotoAlbumRootPath=\\imageserver\photoalbumexports
```

File structure

The export job will place the photo albums in a subfolder to the root path named

```
<site>-<date>-<category>-<PhotoalbumID>
```

where:

<category>	is the category of the photo gallery
<date>	is the date of the photo gallery
<PhotoalbumID>	is the ID of the album
<site>	is the site code of the site

If the <PhotoalbumID> folder already exists, it will be emptied before the exported files are copied to it.

The images are exported as JPG files and meta data is stored in an XML file named

```
<site>-<date>-<category>-<PhotoalbumID>-<PhotoId>
```

where:

<category>	is the category of the photo gallery
<date>	is the date of the photo gallery
<PhotoalbumID>	is the ID of the album
<PhotoId>	is the priority of the image.
<site>	is the site code of the site

The format of the XML file can be controlled by templates.

Object files

The following object files are used for photo album export:

- ExpPhotoAlbums.pbo (optional)
- ExpPhotoAlbum.pbo
- ExpPhotoAlbumItem.pbo

The export object files are part of the default edit templates; however, it is also possible to add user defined object files. These object files must be prefixed with a name (name_). For example:

- CCE_ExpPhotoAlbums.pbo (optional)
- CCE_ExpPhotoAlbum.pbo
- CCE_ExpPhotoAlbumItem.pbo

Using the mentioned syntax, it is possible to have multiple sets of templates for exporting photo albums. Upon export, users determine which templates to use through the **Template set** drop-down in the **Photo albums export** window. The path for user defined templates must exist in **web.dbo.siteconfig** table with the following values:

SITE	SECTION	NAME	VALUE
XX	SETTINGS	AlbumExportTemplates	\\templateserver\xx\AlbumExportTemp\

Where XX is the site ID and Value is the path to the user defined templates. The path is visible to the edit web server.

ExpPhotoAlbum.pbo

This is required. It includes the information for each photo album.

<%albumid%>	The photo album's unique ID.
<%AlbumPictures%>	A placeholder for each exported photo album item (image).
<%category%>	The category in which the photo album is placed.
<%createdate%>	The photo album's creation date.
<%createdby%>	The name of the user who created the photo album.
<%\date%>	The album's publication date.
<%lastmodifiedby%>	The name of the user who last modified the photo album.
<%lastmodifieddate%>	The date on which the photo album was last modified.
<%numberofimages%>	The total number of image in the photo album.
<%published%>	Indicates whether or not the album is published.
<%site%>	The ID of the Site on which the album is placed.
<%\summary%>	The photo album's summary.
<%\title%>	The photo album's title.
<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of ExpPhotoAlbumAccessControlsItem.pbo

ExpPhotoAlbumAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.

<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

ExpPhotoAlbumItem.pbo

This is required. It includes the information about the photo album image. All images are exported in JPG format. The created image files are placed in the same directory as the export file.

<%credit%>	The name of the photographer or creator of the image.
<%imagecontentbase64%>	If enabled, the images are embedded in the XML file and the variable `*<%imagename%*` > will not have any value.
<% imageHeight%>	The height of the image.
<%imagemimetype%>	The image MIME type.
<%imagename%> (created file)	The name of the image file.
<%imageorder%>	The image's ranking in the photo album.
<%Imagewidth%>	The width of the image.
<%\summary%>	The image summary.
<%\title%>	The image title.

ExpPhotoAlbums.pbo

This is optional. If the **ExpPhotoAlbums.pbo** file exists, all the selected albums are exported to a single file. If the *ExpPhotoAlbums.xml *file does not exist, there will be one export file for each photo album.

<%Items%>	Outputs the selected photo albums.

Set picture as desktop background

This function allows the web guest to set an image from the database as his desktop background. You should include the following text to guide the user in this operation.

- Windows XP: Right-click on the photo to the right, and select *Set as desktop background*.
- Macintosh: Click and hold your mouse button down on the photo to the right. When the drop-down menu appears, select *Download Image to Disk*. From the Apple Menu, choose Control Panel > Appearance. Select the **Desktop** tab at the top of the window. Click *Place Picture* and choose the photo you downloaded. If you already have a picture set as your desktop, choose *Remove Picture* first and then click on *Place Picture*. Finally, select *Set Desktop.*

Link

To link to this function, use code from ShowImage.pbo.

Template

The template can be named whatever you like. You specify the template in the link, so be sure that you use the same name in both instances. The example uses **photos_desktop.pbs** in the /misc folder.

Tag

```
<pbs:ptshowimage maximagewidth=800 logo="/graphics/tbdotcom.gif" showtitle=1 size=1
quality=80>
```

Options

Logo	Sets which image should be overlayed on the picture as a logo.
Maximagewidth	Sets the maximum width of the image.
Quality	Sets the image quality.
Showtitle	Makes the system overlay the title on the picture. If no title exists, the image copyright will be used instead.
Size	Enter `size=1` to show a thumbnail of the picture, even if the full picture is loaded from the server.

Watermark images

The images in the system can be watermarked with either a text string or an image. This allows you to put a copyright text on your pictures, for example. You can reference the watermark in the URL of the **Pbcsi.dll** also known as the Image dll.

When using a separate image server, the copyright text must be present in the publicus.ini on the image server. If you are running a shared image server, you must use the option IMMAGGE

Watermark with object files

In the object file that is used to display the picture, you add the option that turns watermarking on.

Text watermark

This function allows you to output text on top of the picture.

Default	To add a text watermark to the image, add `&Title=1` after the image link like this: <img src="<%ImgLink%>&Title=1<%MaxWidth%>">. The system will use the text from the publicus.ini variable ImageCopyright as the watermark.

Option	Enter
	`title=2` to output the text with a black outline. This is done to make the text easier to read if the picture contains white areas. If the variable Image Copyright is not set in the publicus.ini, the title of the picture will be used instead.

Image watermark

To add an image watermark to the image, add **&Logo=**... after the image link like this:

```
<img src="<%ImgLink%>&Logo=/folder/file.gif<%MaxWidth%>">
```

The system will place this GIF in the center of the image. Only GIF files can be used for a watermark.

The control the position of the watermark image, add **&LogoXPos=#** and **&LogoYPost=#** after the image link like this:

```
<img src="<%ImgLink%>&Logo=/folder/file.gif&LogoXPos=#&LogoYPos=#">
```

The X and Y positions are measured in pixels and use positive integers. By default, image watermarks are centered.

Watermark with tags

This can only be done on the gallery page where you use the tag pbs ptshowimage.

This function is deprecated. You should use the object file version instead.

Text watermark

To add a text watermark to the image, **Showtitle=1** inside the image tag like this: `<pbs:ptshowimage showtitle=1 quality=100>`. The system will use the text from the Publicus.ini variable GENERAL section#ImageCopyright as the watermark.

Image watermark

To add an image watermark to the image, add **Logo="/folder/file.gif"** after the image link like this: `<pbs:ptshowimage logo="/graphics/tbdotcom.gif">`. The system will place this GIF in the center of the image.

pbs ptimages

<pbs:ptimages>

The permanent gallery is used to build galleries that are available without any connections to a story. In the Online user interface, these images are administered through the **Misc** menu. Activating this functionality often requires a change to the Publicus.ini.

The gallery tag is also used to build image collections based on the day's published pictures. Please note that SAXOTECH Online also has a story-based photo gallery. This function is usually used when you have too many photos for a story. This is documented in the Article Tags chapter.

This tag requires object files.

Setup

To use the permanent gallery images, you must have Images=1 in the MODULES section section of the Publicus.ini. Please see the Publicus.ini chapter for more information about this setting.

Templates

This function can be used on all templates. See the example template permgall_ovr.pbs.

The tag must be inside table tags like this:

```
<table>
<pbs:ptimages>
</table>
```

Options

Columns	Determines how many columns the picture will span. You can set this option from 1 to 9.

Exclude	Allows you to exclude categories when you use the Today option to show the photos of the day. By default, this option is not used. Enter
	`exclude="NEWS%"`
	to exclude photos from the News category and all its subcategories. You can specify multiple categories to exclude in a comma separated list, like this:
	`exclude="NEWS,SPORTS,ART%"`
	You can use this option with the **Include** option.

Group	Determines which gallery the tag should show. The groups are managed through the editing interface. This option cannot be used with the Today option. This option has no default. To use this option, Specify the group by entering
	`group=GROUPNAME`
	Use
	`group=#CATEGORY#`
	to use the current category ID as group. Use
	`group=#MAINCATEGORY#`
	to use the main category ID for the current category as group.

Include	This option allows you to limit the categories when you use the Today option to show the photos of the day. By default, this option is not used. Enter
	
	`include="NEWS%"`
	
	to show photos from the News category and all its subcategories. You can specify multiple categories to include in a comma separated list, like this:
	
	`include="NEWS,SPORTS,ART%"`
	
	You can use this option with the **Exclude** option.

Maximageheight	Sets the picture height. There is no default setting. You can set this option by entering `Maximageheight=xxx` where xxx is any number between 10 and 9999 pixels.
Maximagewidth	Sets the picture width. There is no default setting. You can set this option by entering `Maximagewidth=xxx` where xxx is any number between 10 and 9999 pixels.
Order	Enter `order=modified` to sort the images with the most recently modified on top.
Quality	Controls picture compression. 1 is the lowest and 100 is the highest. By default, quality is set to 60.
Sortorder	By default the images are output with the oldest image first. By adding the sort option, `Order=modified` the images are sorted with the newest image first.

Today	Determines that the gallery should be built from today's pictures. This option cannot be used with the Group option. The date for which the gallery is shown can be controlled by adding `&Date=yyyymmdd` to the URL. This option will only output the main photo from a story. The photo must be in JPEG format.

ImageItem.pbo

<%articleparams%>	The necessary information to link back to the story from which the picture was taken. This function only works with the **Today** option.The field is used like this: `[%{ArticleParams}<a href="<%pbcs.dll%>/article?<%articleparams%>">Read story»</a>%]`
<%Byline%>	The photo credit.
<%Caption%>	The caption for the picture.
<%ImagePath%>	The path used in the `<img>` tag.
<%Title%>	The title of the picture.
<%StoryTitle%>	The title of the story. Only available when using the "today" option.
Zoom link	The zoom link is built using multiple variables, like this: `<a href="/apps/pbcs.dll/misc?url=/misc/photos_single.pbs &Category=PERMGALL&IDato=<%IDato%>&IKategori=<%IKategori%>&ID=<%ID%>">` You can only change the category in this link.

ImageItemRowEnd.pbo

This file does not have any variables.

ImageItemRowStart.pbo

This file does not have any variables.

Newsletter Tags

The Newsletter function allows you to send out a list of all the new stories for the day with links to the stories on your website.

The server must be set up to send out newsletters for your website. Otherwise, web guests can subscribe but no newsletter will be sent. The server setup can only be done by SAXOTECH.

- Setup newsletters
- Web based management for newsletters
    - SubcatSimpleLink1.pbo
- Newsletter template
    - Newsletter.txt
    - NewsletterAd.txt
    - NewsletterArticle.txt
    - NewsletterCategory.txt
    - NewsletterLinks.txt
    - NewsletterOAMSAdHTML.pbo
    - NewsletterOAMSAdSpot.txt
    - NewsletterSniff.txt
- pbs Newslad
- pbs newsletter_subscriptionform
    - NewsletterActivated.pbo
    - NewsLetterActivatedError.pbo
    - NewsletterRemoved.pbo
    - NewsLetterRemoveError.pbo
    - NewsLetterRemoveMail.pbo
    - NewsLetterSubscribeMail.pbo
    - NewsLetterSubscriptionForm.pbo
- Subscription page with Sniff Search

Setup newsletters

Newsletter.ini file

The file that controls the newsletter is called newsletter.ini. This file must be placed in the objects folder. The file has the following sections:

SETTINGS	This is the header for the Settings section.
	• CharSet - Sets the character set of the email. It is set like this:
	`CharSet=iso-8859-1`
	• Language - Sets the correct language for use in the date, for example. The option is used like this:
	`Language=DEN (Danish)`
	`Language=SWE (Swedish)`
	`Language=GER (German)`
	• ReplyTo - Sets the reply to address for the e-mail. The option is used like this: `ReplyTo=sendereditor@tahoe.com`* Sender - Sets which email address the newsletter is sent from. The option is used like this:
	`Sender=info@domain.com`
	• URL - The URL is used to set the base URL for your site. The option is used like this:
	`URL=http://www.nevadaappeal.com`

SYSTEM	This is the header for System section. It must be present.
	• Antall - Sets the maximum number of stories sent from a single category.
	• DateFormat - Sets the date format. This field uses the standard SAXOTECH Online date formatting options. The option is used like this:
	`DateFormat=dddd, mmmm dd, yyyy`
	• From - Sets the sender of the e-mail. The option is used like this:
	`From=nevedaappeal.com`
	• Sniff categories - By default the Sniff Search looks in all categories for the system. To limit Sniff Search categories, use this option like this:
	`SniffCategories=NEWS,SPORT,...`
	• Subject - Sets the subject of the e-mail. The option is used like this:
	`Subject=Your email newsletter from nevadaappeal.com`

Web based management for newsletters

Web-based management of newsletters lets web guests manage their subscriptions through a webpage. When using web-based management the web guest can choose categories from which to receive stories.

User registration

The web guest must be a registered user of your site. This is because the newsletter uses the same database as the White Pages for user registration.

User registration is described in the section User Registration.

Setup

The setup is done using the SAXOTECH Online user interface. For more information, see the *SAXOTECH Online User Manual.*

You should create a category called NLETTER in which you set **Link type** to **subscribe** and **registration** to either **Simple** or **Advanced**. This affects which registration template the web guest will see.

Link to the page

You can link to the page using the ft tag. It will use the link type from the category to create the correct link. If you are going to link to the page manually, use the following link:

```
/apps/pbcs.dll/subscribe?Category=NLETTER
```

Template

The template is called **gen_subscribe.pbs**. You must create a template specifically for the category. If the category you have created is called NLETTER, call the template **nletter_subscribe.pbs**. If you call the template gen_subscribe, web guests can bypass the user registration screen by putting another category name in the URL.

Tags and fields

The following tags and fields are available on the subscription page. Please see the example file **nletter_subscribe.pbs** in the sample template zip file.

Active	Determines whether the subscription function is active. `<input type="hidden" name="Active" value="<pbs:#active#>">`
Category subscribed	This tag outputs the value of the category to which the web guest has subscribed. `<input type="checkbox" name="Category" value="BUSINESS" <pbs:#CB_BUSINESS#>> Business `
First name	Outputs the subscriber's first name. `<pbs:#fname#>`
HTML format	Outputs the value checked if the subscriber has selected to receive HTML mail. `<input type="checkbox" name="HTML" value=1 <pbs:#ReceiveHTML#>>`
Received letters	This tag shows how many newsletter the subscriber has received. `<pbs:#ReceivedLetters#>`
Surname	Outputs the subscriber's surname. `<pbs:#sname#>`

Basic subscription page

Form code	The form must begin with:
	
	`<form name="selections" action="<pbs:prog>/subscribe?UserReg=1" method="POST">`
	
	and must end with:
	
	`</form>`

JavaScript	The JavaScript present in the example code must be on the page.
Listing categories	You must put checkboxes for the available categories in the form. Each checkbox must have the name Category, and the value must be the category ID for the desired category. `<input type="checkbox" name="Category" value="NEWS" <pbs:#CB_NEWS#>>News  <input type="checkbox" name="Category" value="SPORTS" <pbs:#CB_SPORTS#>>Sports  <input type="checkbox" name="Category" value="OPINION" <pbs:#CB_OPINION#>>Opinion ` The pbs tag outputs the checkbox value when a subscription is edited. This tag must be present for each category.
Select/unselect all	The following code is needed if you want a link to subscribe/unsubscribe to all subcategories of a category: `<a href="javascript:SelectAll(true, 'NEWS');">Select all</a> - <a href="javascript:SelectAll(false, 'NEWS');">Unselect all</a> ` You must replace NEWS with the category ID of the category to which the links should apply.

Subscribe/unsubscribe	The following piece of code is needed to output the links that allows the user to subscribe or unsubscribe: `<a href="javascript:doSubmit(0);">Subscribe</a> - <a href="javascript:doSubmit(1);">Unsubscribe</a>`
Unsubscribe field	The form must contain: `<input type="hidden" value="0" name="unsubscribe">` Without this code, the form will not work.

Listing Subcategories

The following code can be used to output the checkboxes for the categories. This code is used to automatically output the checkboxes for all subcategories. This can also be done manually:

```
<pbs:subcategories maincategory="NEWS" class=1 force exclude="NEWS13">
```

This code uses the subcategory function.

Class	The class that determines which object file will be used.
Exclude	You can exclude categories from the listing by entering their category IDs here.
Force	Forces the listing of all subcategories. It is needed to output the subcategories without stories on today's date.
Maincategory	The main category for which you want to list subcategories.

SubcatSimpleLink1.pbo

The subcategory in the example code uses SubcatSimpleLink1.pbo. It must contain the following codes. Please see the example file on how to use these variables.

<%Categoryid%>	This field must be placed inside the value field of the input checkbox.
<%Categoryname%>	Outputs the name of the category.
<%Checked%>	This variable will mark the checkbox if the web guest has subscribed to the category.

Example

```
<input type="checkbox" name="Category" value="<%CategoryID%>" <%Checked%>>
<%CategoryName%><br>
```

Newsletter template

Newsletter template files are used when sending the email to the subscriber. The HTML versions of the templates have HTML added before .txt, so Newsletter.txt becomes **NewsletterHTML.txt**.
All templates must be put in the **objects** folder.

> ⚠ It is important that you keep your lines under 60 characters in order for them to be handled correctly by email servers.

Newsletter.txt

The main file is called **newsletter.txt**. The file supports the following variables:

<%Ad1%>... <%Ad99%>	These fields will output the ads defined in the editorial system.
<%Ads%>	Outputs the ads for the newsletter.
<%Articles%>	Outputs the stories the subscriber has selected to receive.
<%Links%>	Outputs the links for the newsletter.
<%LongDate%>	Outputs the date the e-mail was generated.
<%UnsubscribeURL%>	Contains the link to the unsubscribe page.

NewsletterAd.txt

This file is output for every ad.

<%AdText%>	The ad text.
<%AdURL%>	The link to the ad.
<%HTMLAdText%>	The HTML version of the ad text.
<%Title%>	The ad heading.

NewsletterArticle.txt

This file is called **NewsletterArticle.txt** and it is output for each story. It must be placed in the objects folder.

<%Category%>	The name of the category.
<%\Date%>	The story date in ISO format.
<%ImgLink%>	The link to the main photo attached to the story.
<%Leadin%>	The leadin for the story. To avoid the e-mail server destroying the email, you should add an extra space after each period. This example also shows how you can crop your leadin to a maximum of 300 characters. `<%Leadin$w(1,300)$r(".=. ")%>`
<%Link%>	The link to the story.
<%\Title%>	The title of the story.
<%SniffMatch%>	The Sniff word matched for the story. This field is only available on stories that have been output using Sniff words.

Special variables when using the keyword group file:

<%Category%>	The name of the category.

<%CategoryID%>	The category ID.
<%Super%>	The story kicker.
<%Text%>	The story's first text block.

Predefined **<newsl:ad>** variables:

If **nlsort** is not set, the variables **AdCategory** and **AdProfile** will be set corresponding to **NewsLetterHTMLCategory.txt**.

NewsletterCategory.txt

This file is called **NewsletterCategory.txt** and it is output for each category to which web guest has subscribed. It must be placed in the objects folder.

<%Articles%>	Outputs the stories in the category.
<%Category%>	Outputs the name of the category.

Predefined **<newsl:ad>** variables:

AdCategory	Corresponds to **<%Category%>** if the **nlsort** switch is not set.
AdProfile	If a profile corresponding to the category has been used to select stories it will be used in the tag. This means that the **nlsort** switch is not set, the *profiles *switch is set, the current category has an associated profile, and keywords are not used for the selection.

NewsletterLinks.txt

The file is output when the field **<%links%>** is used in the mail file. Links are added to the newsletter. The links are selected by a command line switch when the program that sends newsletters is run.

<%Description%>	The description of the link.
<%\Title%>	The title for the link.
<%URL%>	The URL for the link.

NewsletterOAMSAdHTML.pbo

Controls the way OAMS banners are placed in the composed list.

<%HTMLOAMSAd%>	The generated OAMS banner.

NewsletterOAMSAdSpot.txt

Controls the way an OAMS ad is called. The available variables are the same as for the **newsl:ad** tag described above.

<%AdLabel%>	The OAMS ad label, if applicable.
<%Categoryk%>	The category.
<%IDN%>	The spot identifier.
<%NoLink%>	Determines whether the link is included.
<%PageId%>	The Page ID.
<%Profile%>	The profile, if applicable.
<%Server%>	A reference to the OAMS server.
<%Site%>	The site.
<%Spot%>	The spot.
<%SubLabel%>	The sublabel, if applicable.
<%Subsite%>	The subsite.

NewsletterSniff.txt

<%IsFirstItem%>	Is 1 when processing the first sniff item.
<%Leadin%>	The leadin for the story. To avoid the e-mail server destroying the e-mail, you should add an extra space after each period. This example also shows how you can crop your leadin to a maximum of 300 characters. `<%Leadin$w(1,300)$r(".=. ")%>`
<%Link%>	The link to the story.
<%SniffMatch%>	The Sniff word matched for the story.
<%\Title%>	The title of the story.

Predefined **<newsl:ad>** variables:

If **nlsort** is not set, the variables **AdCategory** and **AdProfile** will be set corresponding to **NewsLetterHTMLCategory.txt**.

pbs Newslad

<newsl:ad>

It is possible to put OAMS ads into newsletters using **newsl:ad** tags in the HTML versions of the templates. For many of the variables, default values can be defined in the **newsletter.ini** in the **ADX** section. In some cases, some of the variables are predefined. Specific instances are described for each template. The predefined values can be overridden in the tag definition.

AdCategory	The category to use on the spot. Ini file ADX variable: **DefaultCategory**. Default value: `-` but the value should be defined.
AdLabel	Instead of using category or profile use a direct OAMS label. Ini file ADX variable: **DefaultLabel**. Default value: `-` (no label).
AdPageID	An ID common to all OAMS references in the newsletter. Must be unique on the OAMS server. Default: A random ID is generated.
AdProfile	The profile to use on the spot. Ini file ADX variable: **DefaultProfile**. Default value: `-1` (no profile).
AdServer	Name of the server that should handle OAMS calls. Ini file ADX variable: **Server**.
AdSite	The site code used in the OAMS calls.Ini file ADX variable: **Site**. Default: The site defined by the `-v` switch.
AdSubLabel	The ad sublabel. Ini file ADX variable: **DefaultSubLabel** Default: No sublabel.
AdSubSite	The subsite used in the OAMS calls. Ini file ADX variable: **SubSite**. Default value: `001`
IDN	An ID identifying the individual hole. Must be unique on the OAMS server. Default: A random ID is generated.
NoLink	If this is 1, no link will be generated for the OAMS ad and only the banner will be shown. Ini file ADX variable: **NoOamsLinks**. Default value: `0`
Spot	The spot to replace the newsl:ad tag. Ini file ADX variable: **DefaultSpot** Default value: `-` but the value should be defined.

pbs newsletter_subscriptionform

<pbs:newsletter_subscriptionform>

The email based management is a simpler version of the subscription form. This version allows the web guest to enter an email address without registering. An email is sent to the web guest when he subscribes or unsubscribes. This email contains a link he must click to confirm his subscription status. When this option is used for managing newsletters, the web guest cannot change the categories to which he subscribes.

Setup

NewsletterMailID in Publicus.ini sets the sender name for the confirmation emails.

Template

This function can be used on all templates. See the example file Newsletter_ovr.pbs.

NewsletterActivated.pbo

Used when the web guest subscribes to the newsletter successfully. This file has no fields.

NewsLetterActivatedError.pbo

Shown if there is an error when the web guest tries to subscribe to the newsletter. This file contains no fields.

NewsletterRemoved.pbo

Used when the web guest unsubscribes successfully. This file has no fields.

NewsLetterRemoveError.pbo

Shown if there is an error when the web guest tries to unsubscribe to the newsletter. This file contains no fields.

NewsLetterRemoveMail.pbo

Used to send the confirmation email to a web guest who is unsubscribing.

<%url%>	The URL the user must click to confirm the unsubscription.

NewsLetterSubscribeMail.pbo

Used to send the confirmation e-mail to the subscriber.

<%url%>	Contains the URL the user must click to confirm the subscription.

NewsLetterSubscriptionForm.pbo

The NewsletterSubscriptionForm.pbo object file describes the form that users use to subscribe to newsletters.

An example subscription form frame

```
<script>
  function SaveEmail() {
    newsletterform.Remove.value="0";
    newsletterform.Subject.value="Subscribe to the newsletter";
    newsletterform.submit();
  }

  function RemoveEmail() {
    newsletterform.Remove.value="1";
    newsletterform.Subject.value="Unsubscribe to the newsletter";
    newsletterform.submit();
  }
</script>

<form name="newsletterform" action="<%pbcs.dll%>/savenewsletteruser" method="POST">
  Form fields goes here.
</form>

<a href="javascript:SaveEmail();">Subscribe</a><br>
<a href="javascript:RemoveEmail();">Unsubscribe</a><br>
```

Available form fields

Add	This hidden field can be used for telling if the categories in the **categories** field are to be added (=1) to the list of subscribed categories for the email address or if they are to replace (=0) the list. Default is replace (=0). `<input type="hidden" name="Add" value="1">`
Categories	This hidden field is used to set the categories to which the web guest subscribes. `<input type="hidden" name="Categories" value="NEWS01,NEWS02">`
Email	Captures the subscriber's e-mail address. `<input name="Email" size="30" />`

HTML	These radio buttons enable the subscriber to choose between HTML and plain text newsletters.
	
	`<input type="radio" name="HTML" value="0" checked>`
	
	Plain text
	
	`<input type="radio" name="HTML" value="1">`
	
	HTML mail

RedirectURL	This hidden field is used to set the page to which the subscriber will be directed after the form is processed. `<input type="hidden" name="RedirectUrl" value="<%pbcs.dll%>/misc?url=/mal/NewsLetterMailSent.pbs">`
Remove	This hidden field is used to control whether the web guest is subscribing or unsubscribing. This field is filled in by the JavaScript. `<input type="hidden" name="Remove">`
Subject	This hidden field is used to set the subject of the e-mail. This field is filled in by the JavaScript. `<input type="hidden" name="Subject">`

Subscription page with Sniff Search

This subscription page is a more advanced version, which allows the web guest to select stories based on keywords. This is an extension of the basic subscription feature.

Sniff Search

Sniff search is a function that allows a web guest to receive links to all stories with a specific keyword. This functionality can allow Bob's Widget Company to receive all stories in which widgets are mentioned.

Types

The added search word can be one of three types.

Containing	A word in the story must contain the word. These words are marked with [] in the list of words. If the sniff word is tree, it would match *tree, treehugger* and *street*.
Matching	The word must match a word in the story exactly. If the sniff word is *tr ee*, it would match tree, but not *treehugger* or *street*.
Starting with	A word in the story must begin with the selected word. These words are marked with <> in the list of words. If the sniff word is *tree*, it would match *treehugger* and *tree*.

Template

This function also uses the _**subscribe.pbs** template. See the example template Sniff_Subscribe.pbs.

Fields

JavaScript	This version of the form requires that all the JavaScripts in the example are present. You should cut and paste the example form and work from there.
Sniff	Used to store the selected sniff words. `<input type="hidden" name="sniff" value="<pbs:#sniff#>">`

Searching

SAXOTECH Online provides the ability to search the stories, forum messages, CCE objects and images in the system. The search functionality has two parts:

- The search form, where the web guest inputs the query.
- The search result page that displays the result.
  Searching CCE objects is also described in the CCE chapter.
- Image search
  - pbs imagesearchresult
    - SearchResultImage.pbo
    - SearchResultImageItem.pbo
    - SearchResultImageRow.pbo
    - ImageSearchError.pbo
    - SearchResultImageItemAccessControlsItem.pbo
    - SearchResultImageItemAccessControls.pbo
- Result page
  - Building the page without object files
    - pbs search_criteria
    - pbs search_hits
    - pbs search_next_link
    - pbs search_previous_link
    - pbs search_totalhits
    - pbs searchresult without object files
  - pbs searchresult
    - StorySearchNavigatorItem.pbo
    - ImageSearchNavigator.pbo
    - ImageSearchNavigatorItem.pbo
    - SearchError.pbo
    - SearchNavigator.pbo
    - SearchNavigatorItem.pbo
    - SearchResult.pbo
    - SearchResultItem.pbo
    - StorySearchNavigator.pbo
    - SearchNavigatorFilter.pbo
    - SearchNavigatorFilterItem.pbo
    - SearchResultItemByline.pbo
    - SearchResultItemTaxonomyItemPathItem.pbo
    - SearchResultItemTaxonomy.pbo
    - SearchResultItemTaxonomyItem.pbo
    - SearchResultItemAccessControlsItem.pbo
    - SearchResultItemAccessControls.pbo
- Search form
- Search other websites
- Search syntax

Image search

SAXOTECH Online search functionality has two parts regarding free text searching:

- The search form where the web guest inputs the query.
- The search result page that displays the result.

Image search allows for searching all images related to stories and photo galleries. Image search will search stories in the following text fields found in the Index Profile: title, net title, subtitle, byline, body, image texts, image captions.

Template

Image search results are shown using the <pbs:imagesearchresult> tag in the **gen_imagesearch.pbs** template. It is also possible to specify a template using the keyword TEMPLATE=TemplateName in the search action.

Image search uses the **imagesearch** action.

Options

StartDate	This is the first date from which the system will select images. This date field must be in the YYYYMMDD format. It can be both visible and hidden.
EndDate	This is the last date from which the system will select images.

BuildNavigators	If BuildNavigators has value="1", navigators defined in the NavigatorList in <pbs:imagesearchresult> will be built.
Criteria	The field for entering the search criteria.

SearchCategory	Determines which categories will be searched. You can set the search category in the following ways:
	
	`<input type="hidden" name="SearchCategory" value="%">`
	
	`<input type="hidden" name="SearchCategory" value="Categoryid">`
	
	`<input type="hidden" name="SearchCategory" value="Categoryid1;Categoryid2">`
	
	`<input type="hidden" name="SearchCategory" value="Categoryid%">`
	
	`<input type="hidden" name="SearchCategory" value="Categoryid[1-3]">`
	
	You cannot set the search category using multiple categories and wildcards.

ExcludeCat	The same rules as for setting search categories apply when excluding categories.
SearchProfile	You can limit the search to a profile.
IncludeStatus	Add this field to search stories that are not published.

Example showing the use of the imagesearch action:

```
<form method="post" action="imagesearch?Category=SEARCH">
<input type="hidden" name="EndDate" value="20060606">
<input type="hidden" name="StartDate" value="19600101">
<input type="hidden" name="BuildNavigators" value="1">
<input type="hidden" name="ExcludeCat" value="ARTICLES">
<input type="checkbox" name="IncludeStatus" value="7">
<input type="text" name="Criteria" style="width : 95px" size="8">
<input type=submit name=Go VALUE="Go">
</form>
```

pbs imagesearchresult

<pbs:imagesearchresult>

Options

Navigatorlist	A list of navigators to include in the search result. `Category, Date`
ObjectClass	Specifies the object class used.
Sites	Used when you are combining multiple sites in a portal. If you are not searching multiple sites you do not need to add this option. To use this option, enter the site codes in a comma separated list.
ImageColumns	Specifies the number of images shown in one row. The default value is "1".
ResultsOnPage	This option is used for controlling how many results should be shown per page. The default value is "12".
SortField	Sorts the result according to the field specified. The default is the publication date. A list of possible sort fields is provided in Publicus.ini.
SortDirection	Determines whether the sort is ascending or descending. The default is descending.

Example showing the use of `<pbs:imagesearchresult>`:

```
<pbs:imagesearchresult Showref=1 sites="DA" ResultsOnPage=6 navigatorlist="Keywords,
Locations" imagecolumns=3 objectclass=1 sortfield="news">
```

SearchResultImage.pbo

<%SearchCriteria%>	The text of the search query.
<%SearchPreviouslink%>	The link to previous page of results.
<%SearchNextlink%>	The link to the next page of results.
<%TotalHits%>	The total number of images found.
<%SearchHitRange%>	The value for *from* - *to* hits the page is showing.
<%SearchResult%>	Where the results defined in SearchResultImageItem.pbo and SearchResultImageRow.pbo will be listed.
<%SearchNavigator%>	Where the navigators will be placed.

SearchResultImageItem.pbo

This pbo file is needed to define each result item. The items returned can either be images belonging to stories, or images found in photo albums.

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of SearchResultImageItemAccessControls.pbo
<%DocumentNumber%>	The story number or photo album number depending on the type of result.
<%DocumentByline%>	The byline of stories or photo albums depending on the type of result.
<%DocumentTitle%>	The title of stories or photo albums depending on the type of result.

<%DocumentStatus%>	The status of stories or photo albums depending on the type of result. In photo album's status will always be -9.
<%DocumentDate%>	The publication date of stories or photo albums depending on the type of result.
<%DocumentCategory%>	The category of stories or photo albums depending on the type of result.
<%DocumentLink%>	The document link of stories or photo albums depending on the type of result. For stories, this will link to the relevant story. When an image from a photo album is returned, a link to a photo album will be returned. This requires the appropriate templates.
<%ItemCounter%> **From version 6.9.0sr1**	Output item number.
<%Counter%> **From version 6.9.0sr1**	Output counter.
<%DocumentFlowStatus%>	The flow Status
<%IsFromPhotoAlbum%>	Returns "1" if the image is from a photo album. Otherwise it returns "0".
<%DocumentLeadIn%>	The lead-in from stories.
<%ImageSrc%>	The image result storage location (src), independent of image type (story-connected or photo album).
<%ImageCaption%>	The image caption.
<%ImageText%>	The image text.
<%ImageByline%>	The image byline.

SearchResultImageRow.pbo

Defines the number of images shown in a row. The number of images shown can be specified in `<pbs:imagesearchresult>`.

<%RowItems%>	The number of image results specified in the **ImageColumns** option in `<pbs:imagesearchresult>`. The **ImageColumns** option specifies the number of images that should be shown in one row.

ImageSearchError.pbo

This object file is output when an image search fails.

<%ErrorMessage%>	Contains the error message shown in case of error.
<%Search_Criteria%>	Returns the text used to search the image.

SearchResultImageItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

SearchResultImageItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of SearchResultImageItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

Result page

The result page is used displaying the result page of a search.

Template

The template for displaying the result is called **gen_search.pbs**.

You have two options when building a result page:

- If the object files exist, the page will be built using object files.
- You can use the #tag# tags directly in the search result page.

Building the page without object files

You can place the following tags on the result page.

pbs search_criteria

<pbs:#search_criteria#>

This tag outputs the text of the query.

pbs search_hits

<pbs:#search_hits#>

This tag outputs the hits range the web guest is viewing.

pbs search_next_link

<pbs:#search_next_link#>

This tag outputs a link to the next page of hits. The tag will only output a link if there is a next page. The label will be shown anyway.

Label	This option sets the text string that should be used for linking to the next hits.

pbs search_previous_link

<pbs:#search_previous_link#>

This tag outputs a link to the previous page of hits. The tag will only output a link if there is a previous page. The label will be shown anyway.

Label	This option sets the text string that should be used for linking to the previous hits.

pbs search_totalhits

<pbs:#search_totalhits#>

This tag outputs the total number of hits.

pbs searchresult without object files

<pbs:searchresult>

This tag outputs the search result.

Infocolor	This color will be used to show the category and date.

Showref	1 = the lead-in of the story will be shown0 = the lead-in of the story will not be shown.

pbs searchresult

<pbs:searchresult>

If you want to use object files, place the `<pbs:searchresult>` tag on the template. If not, you must use the other tags described in this chapter.

> The system will only use the object files, if both SearchResult.pbo and SearhResultItem.pbo exists.

The <pbs:searchresult> tag has the following options:

Showref	Determines whether summaries will be retrieved from the database. If **Showref=1** then the summaries will be retrieved from the database.If **Showref=0** then the summaries will not be retrieved from the database. If you do not plan to show summaries, you should use this option to gain speed.
Sites	Used when combining multiple sites in a portal. If you are not searching multiple sites, you do not need this option. To use this option, enter the site codes in a comma separated list.
Alltext	Determines that all the text will be shown for each story, instead of only the summary.
Count	Determines how many results will be shown per page. The default value is 10.

SearchType	SearchType can have two values: Forum or Story. If both stories and forum messages should be searched, two
	`<pbs:searchresult>`
	tags must be placed on the template. One with **Searchtype="forum"** and another with **Searchtype="story"**.By default, if no search type is specified, Search Type will default to **Story**.This is an example taken from the gen_search template:
	`<pbs:searchresult showref=1 navigatorlist="Publishdate,Locations, Names, Companies, Rating" searchtype="story">`
	`<pbs:searchresult showref=1 navigatorlist="Publishdate,Locations, Names, Companies, Rating" searchtype="forum">`
	If a search type is specified in both
	`<pbs:searchresult>`
	and in the /search action, the search type used in
	`<pbs:searchresult>`
	will have precedence over the search action.

RatingSearch	Used to search stories by rating. Example:
	`<pbs:searchresult ratingsearch="1,2,3">`
	will return all stories with a rating of 1, 2, or 3.
	`<pbs:searchresult ratingsearch="2">`
	will only return stories with a rating of 2.It is also possible to add the **r atingsearch** parameter through the standard search form using POST variables or GET variables.

ExcludeCategories	Used to exclude specific categories in the result. `<pbs:searchresult ExcludeCategories="ATP,RZ">`
IncludeNodateArt	Makes the system search stories that are placed on the date 99999999. `<pbs:searchresult IncludeNodateArt="1">`
IncludeProfile	Used to include a profile associated to the story category in the result link. This option is only available when using Online Search powered by FAST. Example: `<pbs:searchresult IncludeProfile="1">`
IncludeStatus	Used for including stories that don't have publish status 9. If IncludeStatus is used as a query parameter and also in the tag the one specified in the tag will take precedence.
Useobject	Determines whether or not object files are used. The default setting is 0, indicating that object files are not used. To enable object file usage, the option setting should be 1.
Objectclass	Determines which object class is used. By default, the system will use the files in the default object class. Objectclass can be set from 1 to 999.

Infocolor is documented here: **pbs searchresult without object files**

Facet options

Facets are only available when using Solr search engine for searching. Also see the Type setting in SearchEngine section in publicus.ini for enabling Solr.

WrapFacet	WrapFacets make it possible to choose a set of pre-defined facet (navigator) fields and have the dll translate them to the appropriate fields in Solr. The default is 0. This option works with story and image search.

NavigatorList	Specifies which navigators are shown in the result set. If WrapFacets is enabled the following fields will be available in the navigatorlist option: `Keywords, TaxonomyTree, TaxonomyLeaf, TaxonomyLeafId, Taxonomy, TaxonomyId, Rating, Location, Byline, Date, GeoName, GeoLocation, Channels, NumberImages, Category` If a field that doesn't exist in the pre-defined list is listed the dll will try to make a custom field facet with that name and generate a facet if it exists. If WrapFacets is off Solr field names must be specified in the navigator list.
SortField	Sorts the result according to the field specified. The default is publication date. A list of possible sort fields is provided in Publicus.ini.
SortDirection	Determines whether the result is sorted in ascending or descending order. The default is descending.

This example shows the result tag for a story and a forum search:

```
<pbs:searchresult Showref=1 Sites="DA" Count=10 wrapfacets=1
navigatorlist="keywords,byline,location,Category,TaxonomyLeaf,Taxonomy,saxonote"
sortdirection="ASCENDING">
```

If wrapfacet is 0 you need to know the field names in Solr in order to create facets like the following example:

```
<pbs:searchresult Showref=1 Sites="DA" Count=10 wrapfacets=0
navigatorlist="f_categorydescription, f_category" sortdirection="ASCENDING">
```

This example shows the result tag for image search:

```
<pbs:imagesearchresult Showref=1 Sites="DA" ResultsOnPage=6 imagecolumns=3
sortfield="articlenumber" sortdirection="ASCENDING">
```

StorySearchNavigatorItem.pbo

Used for preparing search in stories.

<%Counter%>	Countains the counter value.
<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	The number of hits related to the current modifier.
<%ModifierValue%>	Returns a pre-made search string that, if chosen, will be used as input to search criteria when performing a new search.
<%NavigatorName%>	The name of the current navigator.

Example:

```
<a href="<pbs:search>&crit=<%ModifierValue%>"><%ModifierName%>(<%ModifierHits%>)</a>
```

ImageSearchNavigator.pbo

Used for preparing search in images.

<%NavigatorName%>	The name of the current image navigator.
<%NavigatorHits%> **Removed if Solr is used as search engine**	Specifies the number of hits in the result set used in the image navigator.
<%Modifiers%>	Holds all elements (modifiers) for one image navigator. All modifiers use the ImageSearchNavigatorItem.pbo file.

ImageSearchNavigatorItem.pbo

Used for preparing search in images.

<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	Returns the number of hits related to the current modifier.
<%ModifierValue%>	Returns a pre-made search string that if chosen should be used as input to search criteria when performing a new search.
<%NavigatorName%>	The name of the current navigator.

Example:

```
<a href="<pbs:search>&crit=<%ModifierValue%>"><%ModifierName%>(<%ModifierHits%>)</a>
```

SearchError.pbo

This object file is output when a search is for less than three characters, or with larger errors like database connection errors.

<%ErrorMessage%>	Contains the error message shown in case of error.

SearchNavigator.pbo

Used for preparing search in forum messages.

<%NavigatorName%>	The name of the current forum navigator.
<%NavigatorHits%> **Removed if Solr is used as search engine**	Specifies the number of hits in the result set used in the forum navigator.
<%Modifiers%>	Holds all elements (modifiers) for one forum navigator. All modifiers use the SearchNavigatorItem.pbo file.

SearchNavigatorItem.pbo

Used for preparing search in forum messages.

<%ModifierName%>	The name of the current modifier.
<%ModifierHits%>	The number of hits related to the current modifier.
<%ModifierValue%>	Returns a pre-made search string that, if chosen, will be used as input to search criteria when performing a new search.
<%NavigatorName%>	The name of the current navigator.

Example:

```
<a href="<pbs:search>&crit=<%ModifierValue%>"><%ModifierName%>(<%ModifierHits%>)</a>
```

SearchResult.pbo

This is the framework for the search result.

This pages will only be parsed if SearchResultItem.pbo are created.

> From version 6.6.0 the variable names listed below should be used.

<%SearchCriteria%>	The string for which the web guest searched. This tag is similar to the tag you use when you do not have object files.
<%SearchNavigatorFilters%>	Holds the current selected navigator filters. To show the filters the following two object files must be available: SearchNavigatorFilter and SearchNavigatorFilterItem. In order for this to work UseNavigatorFilters=1 is required in the [SEARCHENGINE] section of the publicus.ini. Currently the [SEARCHENGINE] section is in the publicus.ini on the servers (not the repo). If this feature is not working for you contact your CSR and reference this documentation.
<%SearchHitRange%>	The value of *from - to* hits the page shows. This tag is similar to the tag you use when you do not have object files.
<%TotalHits%>	The total number of stories found. The system will not return more than 1000 hits. This tag is similar to the tag you use when you do not have object files.
<%SearchPreviousLink%>	The link to the previous page of results. The recommended method to use this criteria is as follows: `<form action="search" method="post"> <div id="Prevsearch">Previous <input type="hidden" name="url" value="<%SearchPreviousLink%>"> <input src="xxxxxxx" alt="search" style="vertical-align: middle;" type="image"> </div> </form>`
<%SearchNavigator%>	Returns navigators for Guided search.
<%SearchNextLink%>	The link to the next page of results. It must be placed inside a <a `href=""`> tag.The recommended method to use this criteria is as follows: `<form action="search" method="post"> <div id="Nextsearch">Next <input type="hidden" name="url" value="<%SearchNextLink%>"> <input src="xxxxxxx" alt="search" style="vertical-align: middle;" type="image"> </div> </form>`

<%SearchResult%>	Where the results will be listed.If, for example, no articles are found, then search result returns the message "No Articles Found." This message can be modified using a regular expression such as the following:
	`[%{Search_result="No articles found."}`
	`Your search returned no results. Please modify your search criteria and try again.`
	`%|%`
	`<%Search_result%>%]`

The variable names listed below must be used if you are running a pre-6.6.0 release.

<%search_criteria%>	The string for which the web guest searched. This tag is similar to the tag you use when you do not have object files.
<%search_hits%>	The value of *from - to* hits the page shows. This tag is similar to the tag you use when you do not have object files.
<%search_totalhits%>	The total number of stories found. The system will not return more than 1000 hits. This tag is similar to the tag you use when you do not have object files.

<%search_previouslink%>	The link to the previous page of results. The recommended method to use this criteria is as follows:
	`<form action="search" method="post">`
	`<div id="Prevsearch">Previous`
	`<input type="hidden" name="url" value="<%search_previouslink%>">`
	`<input src="xxxxxxx" alt="search" style="vertical-align: middle;" type="image"> </div> </form>`

<%search_navigator%>	Returns navigators for Guided search.

<%search_nextlink%>	The link to the next page of results. It must be placed inside a <a `href=""`> tag.The recommended method to use this criteria is as follows:
	`<form action="search" method="post">`
	`<div id="Nextsearch">Next`
	`<input type="hidden" name="url" value="<%search_nextlink%>"> <input src="xxxxxxx" alt="search" style="vertical-align: middle;" type="image"> </div> </form>`

<%Search_result%>	Where the results will be listed.If, for example, no articles are found, then search result returns the message "No Articles Found." This message can be modified using a regular expression such as the following:
	`[%{Search_result="No articles found."}`
	`Your search returned no results. Please modify your search criteria and try again.`
	`%|%`
	`<%Search_result%>%]`

SearchResultItem.pbo

This object file is output for each item the search returns.

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of SearchResultItemAccessControls.pbo
<%ArticleAddress%>	The URL of the website on which the story was published. This is only useful for a portal that wants to link to a page on another site.
<%ArticleCategory%>	The category name in which the story was published.
<%ArticleCategoryId%>	The category ID in which the story was published.
<%ArticleDate%>	The story's date field.
<%ArticleLink%>	The URL for the story.
<%ArticleModifiedDate%>	The story's ModifiedDate field matching the story's <%ISOModifiedDate%>
<%ArticlePublishDate%>	The story's PublishDate field matching the story's <%ISOPublishDate%>
<%ArticleSource%>	The name of the website on which the story was published. This is only useful for a portal that searches multiple sites.
<%Byline%>	Outputs the byline. If using smartbyline, the template: SearchResultItemByline.pbo is used.
<%Byline%>	The byline for the story.
<%ISOArticleDate%>	The story's date field matching the story's <%ISODate%>.
<%ISOArticleModifiedDate%>	The story's ModifiedDate field matching the story's <%ISOModifiedDate%>
<%ISOArticlePublishDate%>	The story's PublishDate field matching the story's <%ISOPublishDate%>
<%LeadIn%>	The lead-in of the story. ⚠ <%LeadIn%> only appears when Showref=1 is selected in the pbs:searchresults tag.
<%MainImageUrl%>	Link to main image on story.
<%Rating%>	The story's rating.
<%Theme%> **From version 7.0.0sr8**	Contains the theme of the story.

<%ItemCounter%> **From version 6.9.0sr1**	Contains counter for all items.
<%Counter%> **From version 6.9.0sr1**	Contains the count number.
<%SEOLabel%> **From version 7.0.0sr11**	Contains the SEO label of the story.
<%Status%>	Outputs different messages based on the status code for the story. Please see Article Status for information about the valid statuses. The field can be used like this: `[%{Status=7}<font color="#ff0000"><b>This is a restricted article</b> </font>%]`
<%SubTitle%>	The story's sub title.
<%TeaserUrl%>	The URL attached to a teaser
<%TeaserUrlTarget%>	Contains target on teaser URL.
<%Title%>	The title of the story.
<%Flow%>	Contains the flow ID.
<%CurrentFlowStatus%>	Contains the current status of the story within a flow using an ID.
<%CurrentSingleSale%>	Contains the single sale Status ID associated to a story.
<%TaxonomyList%> **From version 6.8.0sr23, 6.9.0sr5**	A list of taxonomy words on the item. This is rendered by by SearchResultItemTaxonomy.pbo.

StorySearchNavigator.pbo

Used for preparing search in stories.

<%NavigatorName%>	The name of the current story navigator.
<%NavigatorHits%> **Removed if Solr is used as search engine**	Specifies the number of hits in the result set used in the story navigator.
<%Modifiers%>	Holds all elements (modifiers) for one story navigator. All modifiers use the StorySearchNavigatorItem.pbo file.

SearchNavigatorFilter.pbo

Holds all the selected navigator filters. This template is used for both story and image search.

<%Items%>	All the filter items.

In order for this to work UseNavigatorFilters=1 is required in the [SEARCHENGINE] section of the publicus.ini. Currently the [SEARCHENGINE] section is in the publicus.ini on the servers (not the repo). If this feature is not working for you contact your CSR and reference this documentation.

SearchNavigatorFilterItem.pbo

Used for generating each individual selected filter. This template is used for both story and image search.

<%FilterName%>	The name of the filter item.
<%FilterValue%>	Value used in selected filter.
<%RemoveFilterUrl%>	Remove the selected filter from the search result.

In order for this to work UseNavigatorFilters=1 is required in the [SEARCHENGINE] section of the publicus.ini. Currently the [SEARCHENGINE] section is in the publicus.ini on the servers (not the repo). If this feature is not working for you contact your CSR and reference this

documentation.

SearchResultItemByline.pbo

The byline file operates in different modes depending on whether or not Smart Bylines are enabled. To use the Smart Bylines feature, you must use `SearchResultItemByline.pbo`.

If the Smart Bylines feature is enabled, the byline will show the first name, surname, and e-mail address of the staff member (as entered in the Staff component) selected in the user interface.

Templates example

Here is an example on supporting Smart Byline and regular byline:

```
[%{namecount}
 [%{namecount=1}<%BylineNamePart%>%]
 [%{namecount=2}<%BylineNamePart%>,<%BylineNamePart2%>%]
 [%{namecount=3}<%BylineNamePart%>,<%BylineNamePart2%>,<%BylineNamePart3%>%]
 %|%
 <%Byline%>
%]
```

<%Address1%>	The first part of the address as defined in the staff registry.
<%Address2%>	The second part of the address as defined in the staff registry.
<%Address3%>	The third part of the address as defined in the staff registry.
<%Address4%>	The fourth part of the address as defined in the staff registry.
<%Byline%>	Outputs the names of all authors with the names as links to the e-mail addresses of those authors the system finds in the register. The function also outputs the other lines in the story.
<%BylineEmailPart%>	Contains the author's e-mail address as saved in the staff register. The same functions as for **<%BylineNamePart%>** are available here.
<%BylineNamePart%>	Contains the author's name as registered in the staff register. If you have two names in the byline you also have the field **<%BylineNamePart2%>**. This function supports up to 9 names in the byline.
<%City%>	Used for Smart Bylines. Contains the author's city as saved in the staff register.
<%Counter%>	Used for Smart Bylines. Contains the count of the staff as saved in the staff register.
<%Department%>	Used for Smart Bylines. Contains the author's department as saved in the staff register.
<%Extension%>	Used for Smart Bylines. Contains the author's work phone number extension as saved in the staff register.
<%Fax%>	Used for Smart Bylines. Contains the author's fax number as saved in the staff register.
<%Field_xxx%> **From version 6.9.0sr2**	Extracts extra fields for the staff member. Replace **xxx** with the name of the field. E.g., `<%Field_Facebook%>`
<%Firstname%>	Used for Smart Bylines. Contains the author's first name as saved in the staff register.

<%Group%>	Used for Smart Bylines. Contains the author's group as saved in the staff register.
<%ImgLink%>	Contains the URL of the author's picture. For additional authors use the field **<%ImgLink2%>**. This tag supports up to 9 people.
<%Information%>	Used for Smart Bylines. Contains the author's information as saved in the staff register.
<%IsPrimary%>	Used for Smart Bylines. `IsPrimary=1` indicates that the author is the primary author.
<%LastName%>	Used for smart bylines. Contains the author's surname as saved in the staff register.
<%MobilePhone%>	Used for Smart Bylines. Contains the author's mobile phone number as saved in the staff register.
<%Name%>	Outputs the names of all authors without e-mail links. The function also outputs the other lines in the story.
<%Namecount%>	Contains the number of authors the system has recognized. It can be used like this: `[%{namecount=1}<%BylineNamePart1%>%|%<%BylineNamePart%> and <%BylineNamePart2%>%]`
<%Office%>	Used for Smart Bylines. Contains the author's office as saved in the staff register.
<%PagerNumber%>	Used for Smart Bylines. Contains the author's pager number as saved in the staff register.
<%Position%>	Used for Smart Bylines. Contains the author's position as saved in the staff register.
<%PrivateEmailAddress%>	Used for Smart Bylines. Contains the author's private email address as saved in the staff register.
<%PrivatePhone%>	Used for Smart Bylines. Contains the author's private phone number as saved in the staff register.
<%PrivateWebAddress%>	Used for Smart Bylines. Contains the author's private web address as saved in the staff register.
<%StaffID%>	Contains the value from the staff ID field. This allows you to link to the information page for one author. The field is used like this: `<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>"><%BylineName Part%></a>` Please see pbs staffinfo for more information.
<%Surname%>	Used for Smart Bylines. Contains the author's surname as saved in the staff register.
<%TotalCount%>	Used for Smart Bylines. Contains the total number of authors for the story.

<%WorkEmailbAddress%>	Used for Smart Bylines. Contains the author's work email address as saved in the staff register.
<%WorkPhone%>	Used for Smart Bylines. Contains the author's work phone number as saved in the staff register.
<%WorkWebAddress%>	Used for Smart Bylines. Contains the author's work web address as saved in the staff register.
<%Zipcode%>	Used for Smart Bylines. Contains the author's zip code as saved in the staff register.

SearchResultItemTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

SearchResultItemTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

SearchResultItemTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses SearchResultItemTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

SearchResultItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

SearchResultItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of SearchResultItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

Search form

The search form is where the web guest enters her query.

Most field values can also be coded in the **Searchresult** tag or as parameters in the URL, in addition to being set using fields in the search form.

Example form code

```
<form method="get" action="<pbs:search>">
```

By using the line of code below, you can choose which template should be used to display the search results.

```
<input type="hidden" name="Category" Value="CategoryID" />
```

For example, without this code, the search result will be shown by the **gen_search.pbs** template.

But with this code:

```
<input type="hidden" name="Category" Value="SPORT" />
```

The **SPORT_search.pbs** template is used.

Templates

The search form can be placed on all pages.

Fields

Input field	The field in which web guest enter the search criteria is called Crit. The field is used like this: `<input type="text" name="crit" style="width : 95px" size="8">` This field is used for passing information to the search program. ⚠ The parameter q can also be used for the search criteria. See q below.

q	The field in which web guest enter the search criteria. The field is used like this:
	
	`<input type="text" name="q" style="width : 95px" size="8">`
	
	or as a query parameter like this:
	
	`http://localhost.com/search?q=test`

Start date	This is the first date from which the system will select stories. This date field must be in the YYYYMMDD format. It can be both visible and hidden. You can also use the calendar from the events search for input of this date. You must give the form a name, and change the name of the field in which the date will be entered. `<input type="hidden" name="StartDate" value="19990101">`
End date	This is the last date from which the system will select stories. Please see **Start date** for more information. `<input type="hidden" name="EndDate" value="20100101">`

Daterange	This option takes two dates, separated by a comma. This option is often used with the Datecalc function to search the current month. The option can be used like this:
	
	`<input type="hidden" name="daterange" value="`
	`<pbs:datecalc now firstofmonth days=-1 firstofmonth>,`
	`<pbs:datecalc now firstofmonth days=-1>">`

Include stories without date	Makes the system search stories that are placed on the date 99999999. The option is enabled like this:
	
	`<input type="hidden" name="IncludeNoDateArt" value="1">`
	
	or
	
	`<input type="hidden" name="INDA" value="1">`

Search category	Determines the categories that will be searched. This can be either a hidden field or a drop-down menu. When specifying multiple categories, you must use a semicolon as the separator. • **Setting search category**You can set the search category in the following ways: `<input type="hidden" name="SearchCategory" value="%"> <input type="hidden" name="SearchCategory" value="Categoryid"> <input type="hidden" name="SearchCategory" value="Categoryid1;Categoryid2"> <input type="hidden" name="SearchCategory" value="Categoryid%"> <input type="hidden" name="SearchCategory" value="Categoryid[1-3]">` You cannot use multiple categories and wildcards: `<input type="hidden" name="SearchCategory" value="Categoryid%;Categoryid%">` • **Drop down example**To create a drop-down menu that lets the web guest select a category, you must build a select box like this: `<select name="SearchCategory"> <option value="" SELECTED>All</option> <option value="NEWS01" SELECTED>Local news</option> <option value="NEWS02">Regional news</option> <option value="NEWS03">National news</option> <option value="NEWS04">World news</option> </select>`

Search profile	You can limit the search to a profile. You can build a drop-down for selecting profiles like this:
	
	`<select name="SearchProfile">`
	`<option value="0"`
	`SELECTED>All</option>`
	`<option value="1001">News</option>`
	`<option value="1002">Sports</option>`
	`</select>`
	
	Specifying a nonexistent profile will result in profiles being ignored. You can also set the profile using a hidden field.

Excluding categories from search	You can exclude specific categories from a search. The option is set like this:
	
	`<input type="hidden"`
	`name="ExcludeCategories"`
	`value="Categoryid">`
	`<input type="hidden" name="ExclCat"`
	`value="Categoryid1;Categoryid2">`
	
	The same rules as for setting a search category apply when excluding categories. Please see **Search category**.

Keyword search	This option is used for searching by keyword in the system. It can be used with the Getkeywords function. For more information, please see [pbs GetKeywords.The keyword search is used as in this example:
	
	`<select name="Keyword" size="1">`
	`<option value=" " SELECTED>Select`
	`Topic</option>`
	`<pbs:getkeywords category="NOTICES"`
	`options=1>`
	`</select>`
	
	If you have a lot of keywords, this function will make the page load very slowly.

Restricted stories	You can also search stories that are not published. To do this, add the field **Includestatus.** The field can be used like this:
	
	`<input type="checkbox"`
	`name="Includestatus" value="7">Search`
	`restricted bylines`
	
	Please see Article Status for more information about valid statuses

Facets	In order to use facets, the following option must be included in the search action for stories, forum messages or images:
	
	`<input type="hidden"`
	`name="BuildNavigators" value="1">`

Search other websites

The multisearch function previously described in this section is obsolete. Please use this JavaScript to achieve the same functionality.

This JavaScript allows you to enter sections of your own site to be searched to the select field, in addition to the available search engines. Also, you can add or remove certain search engines, as needed.

This example is taken from a script written for The Royal Gazette. Engine 1 and 9 are local to The Royal Gazette and are handled differently than the search engines.

The URL for the search engines was gathered by performing a search using the search engines and then modifying the search so the script can add the query term to the URL string passed to the individual search engines. Using this method will allow you to add other search engines to this list.

```
<script language=JavaScript>
function EngineCheck() {
var Engine = '';
for (i=0; i<document.form1.engine.options.length; i++) {
if (document.form1.engine[i].selected) {
Engine = document.form1.engine[i].value;
}
}
if (Engine == '1') {
document.form1.action = 'search';
}
if (Engine == '9') {
document.form1.SearchCategory.value = 'MIDOCEAN';
document.form1.action = 'search?Category=MIDOCEAN';
}
if (Engine == '2') {
var Q = document.form1.crit.value;
document.form1.action =
'http://www.google.com/search?sourceid=mozclient&ie=utf-8&oe=utf-8&q=' + Q;
}
if (Engine == '3') {
var Q = document.form1.crit.value;
document.form1.action = 'http://alltheweb.com/search?cat=web&cs=utf8&q=' + Q +
'&rys=0&_sb_lang=pref';
}
if (Engine == '4') {
var Q = document.form1.crit.value;
document.form1.action = 'http://www.altavista.com/web/results?itag=wrx&q=' + Q +
'&kgs=1&kls=0';
}
if (Engine == '5') {
var Q = document.form1.crit.value;
document.form1.action =
'http://search.lycos.com/default.asp?lpv=1&loc=lycoshp&tab=web&query=' + Q +
'&x=0&y=0';
}
if (Engine == '6') {
var Q = document.form1.crit.value;
document.form1.action = 'http://www.hotbot.com/default.asp?query=' + Q +
'&ps=&loc=searchbox&tab=web&provKey=Inktomi&prov=HotBot';
}
if (Engine == '7') {
var Q = document.form1.crit.value;
document.form1.action = 'http://msxml.excite.com/info.xcite/search/web/' + Q;
}
```

```
if (Engine == '8') {
var Q = document.form1.crit.value;
document.form1.action = 'http://search.yahoo.com/search?p=' + Q +
'&sm=Yahoo%21+Search&fr=FP-tab-web-t&toggle=1';
}
document.form1.submit();
}
</script>
<form name="form1" method="POST">
<input type="hidden" name="SearchCategory" value="%">
<input type="text" name="crit" size="10" maxlength="100">
<br>
<select name="engine" style="width: 120px;">
<option value="1" SELECTED>Royal Gazette </option>
<option value="9">Mid Ocean News</option>
<option value="2">Google</option>
<option value="3">All the web</option>
<option value="4">Altavista</option>
<option value="5">Lycos</option>
<option value="6">Hotbot</option>
<option value="7">Excite</option>
<option value="8">Yahoo</option>
</select><br>
```

```
<input type="button" name="Search" value="  Search &raquo; " onClick="EngineCheck();">
</form>
```

Search syntax

The search syntax is used when the web guest is searching on your pages. This information should be present on your website.

The default search operator is AND.

+, AND

+ is used to determine that the word must appear in the story. This option is not necessary on the first word. The default search operator is AND. This means that all words are required in the story.

Usage

```
michael + jordan + basketball
```

This search will return results where all three words appear.

-, NOT

- is used to select words that must not appear in the story.

Usage

```
michael + jordan - basketball
```

This search will return all stories where *michael* and *jordan* are included, but *basketball* is not included.

%, NEAR

% uses NEAR as the logical operator. This means the words must be near each other in the story.

Usage

```
michael % jordan
```

This search will return all stories where *michael* and *jordan* are close to each other.

" "

This option is used for phrase searches.

Usage

```
"Michael Jordan quits"
```

This search will return all stories where these words appear exactly as written.

OR

Uses OR as the logical operator..

Usage

```
Gothenburg or Sweden
```

This search will return all stories with the word *Gothenburg* or the word *Sweden.* |

( )

This option is used to group searches.

Usage

```
vacation (Holland or Netherlands)
```

This search will return all stories where the words *vacation* and *Holland* or the words *vacation* and *Netherlands* appear.

*****

Used to search for partial words.

Usage The search for

```
foo*
```

will return stories with the words *foot, football* and other words starting with *foo.*

#

Contains. Used for field specific searches.

Usage The search for John Denver within byline can be done like this:

```
byline#(John Denver)
```

This should return all stories with the words John Denver included in the byline. In order to get this to work you will need to url encode the # sign (%23)

If you exclude the parenthesis (group) you will get all stories that has John included in the byline and Denver in one of the text fields.

Custom Content Engine

The Custom Content Engine (CCE) enables you to create, manage and present your own custom objects. Such objects are typically structured information, such as weather, stock quotes, sports results, marketplace/classifieds, recipes, reviews, etcetera. It is especially suited to large volumes of structured data as it is your own custom database. CCE Objects may also serve as classifieds content for OAMS.

CCE tags are used to display the CCE objects.

CCE setup

CCE data retrieval tags

CCE search form

CCE Export

- CCEExport.pbo
- CCEExportCustomFieldItem.pbo
- CCEExportDateMapItem.pbo
- CCEExportGroupBy.pbo
- CCEExportImageItem.pbo
- CCEExportItem.pbo
    - CCEExportItemAccessControlsItem.pbo
- CCEExportItemTaxonomywords.pbo
- CCEExportItemTaxonomywordsItem.pbo
- CCEExportItemTaxonomywordsItemPathItem.pbo

CCE tip a friend

pbs cce

- CCE_ModuleShortName.pbo
- CCE_ModuleShortNameGroupHeader.pbo
- CCE_ModuleShortNameItem.pbo
- CCE_ModuleShortNameItemAccessControls.pbo
- CCE_ModuleShortNameItemAccessControlsItem.pbo
- CCE_ModuleShortNameItemEventDateItem.pbo
- CCE_ModuleShortNameItemTaxonomy.pbo
- CCE_ModuleShortNameItemTaxonomyItem.pbo
- CCE_ModuleShortNameItemTaxonomyItemPathItem.pbo
- CCE_ModuleShortNameNavigator.pbo
- CCE_ModuleShortNameNavigatorFilter.pbo
- CCE_ModuleShortNameNavigatorFilterItem.pbo
- CCE_ModuleShortNameNavigatorItem.pbo
- CCE_ModuleShortNamePageLink
- CCE_ModuleShortNamePageLinkItem
- CCE_MPLACE.pbo
- CCE_MPLACEitem.pbo
- CCE-search_ovr.pbs

pbs ccedetails

- CCE_ModuleShortNameDetails.pbo
- CCE_ModuleShortNameDetailsAccessControls.pbo
- CCE_ModuleShortNameDetailsAccessControlsItem.pbo
- CCE_ModuleShortNameDetailsEventDateListItem.pbo
- CCE_ModuleShortNameDetailsTaxonomy.pbo
- CCE_ModuleShortNameDetailsTaxonomyItem.pbo
- CCE_ModuleShortNameDetailsTaxonomyItemPathItem.pbo
- CCE_ModuleShortNamePrintDetail.pbo
- CCE_ModuleShortNameRelated.pbo

pbs ccefields

- CCE_ModuleShortNameFieldsFieldName.pbo
- CCE_ModuleShortNameFieldsFieldNameItem.pbo
- CCE_ModuleShortNameFieldsFieldNameRow.pbo

For examples of live CCE implementations, please see CCE examples.

Possibilities with CCE - A PowerPoint presentation the introduces CCE concepts

CCE Training Document - PDF used during CCE training

CCE Training Video - A recording of a CCE training webinar.

CCE in the User Manual

pbs LastUpdate for CCE

pbs usersubmitform

CCE setup

Publicus.ini settings

To activate the CCE menu in the user interface, you must add the following value to the **MODULES section** of the Publicus.ini:

```
[MODULES]
CCE=1
```

In the CCE section you can add the following values:

```
[CCE]
CCEClassifieds=0
ClassifiedsModule=11
Cache=1
NoMDCache=1
```

CCE components, classes and types

The first thing to do when creating a new solution with CCE is to use the user interface to setup the meta data for the solution.

- Create the component that will be used when looking up template and object files.
- Create classes for the component.
- For each class, you should specify the required custom fields.
- Add types, subtypes and detail types for the component and/or the classes.

Reserved field names

The following names are reserved and cannot be used as extra custom field names:

- Title
- Date
- Text
- Keyword
- Keywords
- Site
- Category
- Type
- ID
- Expires
- CategoryName
- TypeName
- Image
- AttachedLink
- NavigationCategoryName
- NumImages
- SubType
- SubTypeName
- DetailType
- DetailTypeName
- Count

Other setup

Meta data (for example advertiser's address, phone number, e-mail, or other static meta data) related to CCE objects can be setup through the user interface. The information can be shown using the **Relatedinfo** option.

CCE data retrieval tags

You can retrieve the values of posted fields using the following tags. All other hidden fields in the form can be retrieved using the same method.

Type	`<input type="hidden" name="type" value="<pbs:!#Type#>">`
Range	`<input type="hidden" name="range" value="<pbs:!#Range#>">`
Descending	`<input type="hidden" name="descending" value="<pbs:!#Descending#>">`
Defaultdate	`<input type="hidden" name="defaultdate" value="<pbs:!#Defaultdate#>">`
FromDate	`<input type="hidden" name="FromDate" value="<pbs:!#FromDate#>">`
ToDate	`<input type="hidden" name="ToDate" value="<pbs:!#ToDate#>">`

CCE search form

The search form allows web guests to search the CCE Objects.

Template

The search form can be placed on all pages. The result uses **gen_cce_ModuleShortName.pbs**. This page must have the <pbs:cce> tag to show the results.

Form

The tag <pbs:ccefields> is used in some of the fields to extract information from the database.

```
<form name="ccefsearch" method="POST"
action="<pbs:prog>/cce?Module=2&Category=<pbs:categoryid>">
```

The form code and hidden fields in the example code must be present. If this code is missing, the search form will not work.

Also, you must add the variable **<%SearchString%>** to the submit form for the link to the next page to return the same search string as the original search. **<%SearchString%>** contains the actual search string.

Fields (Single field Searchs)

Almost all options that you can use in the tag can also be used as a field in the form. They are either set using hidden fields, or as a field where the web guest can select the value. The fields are documented with the <pbs:cce> tag.

```
<input type="text" name="TitleSearch" />
```

```
<input type="text" name="TextSearch" />
```

In order to search custom fields, you cannot just use the name of the field. You must append "search" to the end of the custom field name, as in the following example for a custom field called "cost":

```
<input type="text" name="costSearch" />
```

```
<input type="text" name="citySearch" />
```

Fields (Multi field Searchs)

All field can use multi search, the only thing you need todo this to insert the word multi into your name. Like this:

```
<input type="text" name="*customfield*multisearch" />
```

Daterange

```
<input type="hidden" id="daterange" name="daterange" value="20100621-20100621" />
```

Type

```
<input type="hidden" id="Type" name="Type" value="Sport" />
```

Module

```
<input type="hidden" id="Module" name="Module" value="23" />
```

Kategori

```
<input type="hidden" name="*customfield*multisearch" value="Culture,National" />
```

Remember to insert this on your pbs cce tag:

```
<pbs:cce fields="kategori">
```

Exact search

```
<input type="hidden" name="*customfield*SearchExact" value="1">
```

Free text search

You can also use free text search on the CCE objects. Use the following fields:

```
<input type="text" name="Search" value="<pbs:!#Search#>" size="20">
```

This field is used for free text searches. The tag **<pbs:!#Search#>** is used for outputting the user-entered text on the result page.

This example wil find all cce objects where the word "ford" is mentioned in a field.

```
<pbs:cce module=4 class=41 search="ford" sortfield="pubdate">
```

If you don't insert text in the free text input field. The CCE system will use SQL searchs instead. You need to tell CCE what fields / custom fields the SQL searchs will look into. (With fields="X")

```
<pbs:cce execute=1 fields="Name,City,Adress" module=1 class=11 count=10 days=60
fromdate="20100101">
```

More examples

Below are some examples on how you can search the CCE objects.

From/to search
You can also search from and to values. For this to work, you must define two fields:

From:

```
<input type="text" name="fromDocketNumberSearch"
value="<pbs:!#fromDocketNumberSearch#>" size="8">
```

To:

```
<input type="text" name="toDocketNumberSearch"
value="<pbs:!#toDocketNumberSearch#>" size="8">
```

This will allow you to search any value between the from and to values. This can also be used with dates, so you can make a form for searching from/to date. When this function is used, you must remove `Single=1` from the `<pbs:cce>` tag.

Selecting type

```
<select name="Type" style="width: 330px;">
<pbs:!ccefields module=1 class=3 field="Type" optionlist=1 any=1>
</select>
```

Used to select a type. Use the ccefields tag to list all categories for the class.

Online Search powered by Solr in CCE objects

Online Search in CCE can use *near, and, or* and *not* operators to build complex search expressions.

The search parameter is used to set the search expression. See Search syntax.

The SortField parameter is used to set to the sort field. This search can be used in combination with the Module, Class, Range, Fromdate, Todate and Locked parameters.

CCE Export

It is possible to export CCE objects from the edit interface. The following setting must be added to GenereIt in Publicus.ini:

```
CCEExport=1
```

It is possible to use both predefined templates found under templates/CCE or specify a folder with custom templates. Two settings are available the SiteConfig table:
'CCEExportTemplates' have been added to 'SETTINGS' and specifies a folder from where custom templates are loaded.
'CCEExportSavePath' in section 'SETTINGS' specifies where files should be saved.

All object files listed below are required to export CCE objects.

CCEExport.pbo

Contains the framework that is output once. This file is required.

<%items%>	All items exported

CCEExportCustomFieldItem.pbo

Contains the formatting for each custom field for the exported object.

<%FieldName%>	Field name of custom field
<%DisplayName%>	Display name of custom field
<%FieldValue%>	Field value of custom field
<%Type%>	Type of custom field

CCEExportDateMapItem.pbo

Displays the occurrence date of each exported object

<%Date%>	One recurrence date of object
<%FromTime%> **From version 6.9.0sr1**	Contains an ISO formatted datetime of the from time of the event occurrence.
<%ToTime%> **From version 6.9.0sr1**	Contains an ISO formatted datetime of the to time of the event occurrence.

Example:

```
<occurrence value="<%Date%>" fromtime="<%FromTime$d("hh:nn")%>"
[%{ToTime}totime="<%ToTime$d("hh:nn")%>"%] />
```

CCEExportGroupBy.pbo

Outputs the framework for the grouping mechanism in CCE Export.

<%Group%>	Group value. If export is grouped by date, this show the date used for grouping
<%Items%>	All CCE objects found within this group

CCEExportImageItem.pbo

This file contains the formatting for each image or logo that is exported. This file is required.

<%LogoName%>	The name of the logo file. If base64 is selected this will be empty.
<%LogoBase64%>	Contains the base64 encoded logo.
<%ImageName%>	The name of the image file. If base64 is selected this will be empty.
<%ImageBase64%>	Contains the base64 encoded image.

```
[%{logoname|logobase64}
  <logo>
    [%{logobase64}
      <base64>
        <%logobase64%>
      </base64>
    %|%
      <name>
        <%logoname%>
      </name>
    %]
  </logo>
%]
[%{imagename|imagebase64}
  <image>
    [%{imagebase64}
      <base64>
        <%imagebase64%>
      </base64>
    %|%
      <name>
        <%imagename%>
      </name>
    %]
  </image>
%]
```

CCEExportItem.pbo

This file contains the formatting for each object that is exported. This file is required.

<%Site%>	Site code of exported object
<%Module%>	Module ID of exported object
<%Class%>	Class ID of exported object
<%CCEObjectID%>	Object ID of exported object
<%Date%>	Date of exported object
<%CCEType%>	Type ID of exported object
<%Title%>	Title of exported object

<%CCEText%>	Text of exported object
<%Expires%>	Expires time of exported object
<%GUID%>	GUID of exported object
<%OriginalID%>	Original ID of exported object
<%Images%>	Output logo or images exported object.
<%Latitude%>	Latitude of exported object
<%Longitude%>	Longitude ID of exported object
<%CustomFields%>	Holds all custom fields of exported object. See CCEExportCustomFieldItem.pbo
<%Customfield_FieldName%>	Output the value of one custom field. This means that it is possible to output the value of all customs fields without using CCEExportCustomFieldItem.pbo. However users must specify each custom field variable.
<%EventDateMaps%>	Holds all recurrence dates if exported object is a CCE recurrence object. See ModuleType variable.
<%ModuleType%>	Module type of exported object. 0 denotes an object without recurrence and 1 is an object with recurrence.
<%TaxonomyWords%> **From versions 6.9.0sr26, 7.0.0sr1**	Outputs the taxonomywords on the exported CCE object, rendered by CCEExportItemTaxonomywords.pbo
<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of CCEExportItemAccessControlsItem.pbo

⚠ The following variables are only available if ModuleType is 1 (CCE recurrence object):

Item	Description
<%EventDate%>	First coming event date found on a recurrence object.
<%FromDate%>	Start date of event
<%ToDate%>	End date of event
<%FromTime%>	Start time of event
<%EventFromTime%>	Start time of event (ISO format)
<%EventToTime%>	End time of event (ISO format)
<%PatternType%>	Pattern type of recurrence object
<%PatternSubType%>	Sub pattern type of recurrence object
<%Jump1%>	Jump1 of recurrence object
<%Jump2%>	Jump2 of recurrence object
<%ActSun%>	(Sunday) days when recurrence object is active
<%ActMon%>	(Monday) days when recurrence object is active
<%ActTue%>	(Tuesday) days when recurrence object is active
<%ActWed%>	(Wednesday) days when recurrence object is active
<%ActThu%>	(Thursday) days when recurrence object is active
<%ActFri%>	(Friday) days when recurrence object is active
<%ActSat%>	(Saturday) days when recurrence object is active
<%ActThu%>	(Thursday) days when recurrence object is active

<%ActFri%>	(Friday) days when recurrence object is active
<%ActSat%>	(Saturday) days when recurrence object is active

CCEExportItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

CCEExportItemTaxonomywords.pbo

Represents the list of taxonomy words associated with the exported CCE object.

The following variables are availabe:

<%Items%>	Contains the taxonomy words, each rendered by CCEExportItemTaxonomywordsItem.pbo

Example:

```
[%<taxonomywords>
    <%Items%>
  </taxonomywords>%]
```

CCEExportItemTaxonomywordsItem.pbo

Represents a taxonomy word associated with the exported CCE object.

The following variables are available:

<%Description%>	Holds the displayname of the taxonomy word.
<%ID%>	Holds the id of the taxonomy word.
<%Path%>	Holds the full taxonomy tree path of the taxonomy word using '>' as delimiter
<%PathItems%>	Contains the parent taxonomy words, rendered by CCEExportItemTaxonomywordsItemPathItem.pbo (can be used for creating own tree path).
<%Url%>	Holds the Url of the taxonomy word.

Example:

```
<taxonomyword id="<%id%>">
  <displayname><![CDATA[<%<%Description%>]]></displayname>
[%<url><![CDATA[<%Url%>]]></url>%]
  <treelocation>
    <path><![CDATA[<%Path%>]]></path>
  [%<detailedpath>
      <%PathItems%>
    </detailedpath>%]
  </treelocation>
</taxonomyword>
```

CCEExportItemTaxonomywordsItemPathItem.pbo

Represents one of the ancestors of the current taxonomy word associated with the exported CCE object.

<%Parent%>	Holds the taxonomy word's display name
<%IsFirst%>	Outputs whether the taxonomy word is the first in the generated path, i.e., is a root taxonomy word (=1) or not (=0).
<%IsLast%>	Outputs whether the taxonomy word is the last in the generated path, i.e., is the current taxonomy word itself (=1) or not (=0).

Example:

```
<parent isroot="<%IsFirst%>" isself="<%IsLast%>">
  <displayname><![CDATA[<%<%Parent%>]]></displayname>
</parent>
```

CCE tip a friend

This uses CCE templates for the message. Sending a message uses the file **CCE_<ModuleShortName>TipsText.pbo** or **CCE_<ModuleShortName>TipsTextHtml.pbo**.

The CCE send tip form will use **CCE_<ModuleShortName>tip.pbs** if it is present. Otherwise, it will use the **gen_tip.pbs** template.

> There is a setting added in publicus.ini (FallbacktoClassifiedsTemplates=0 in the CCE section). If this setting is OFF, the PrintLink, TipLink, ImageLink, Logolink and DLink will be generated like /apps/pbcs.dll/cce_tips?.. instead of /apps/pbcs.dll/cce_classifieds?..
>
> Also, if the category exists the template name should be changed from CategoryIDCCE_CCEModuleShortNameTip.pbs to CategoryID_CCE_CCEModuleShortNameTip.pbs (Note an underscore is added after categoryID).

pbs cce

<pbs:cce>

The main listing is used to give the overview of CCE objects for the given module For example:

```
<pbs:cce module=4 class=41 fields="Temperature,Condition">
```

The template usually has a search form, enabling web guests to find information.

Templates

The module's short name is used when looking up the main listing template.

The default template to display the list of CCE objects for a module is called **gen_cce_ModuleShortName.pbs**

The template for displaying the object details is called **gen_cced_ModuleShortName.pbs**

The template for displaying the print version is called **gen_ccedp_ModuleShortName.pbs**

Example
If you have a *Weather Data* module, with the short name *weather,* the following templates are used:

gen_cce_weather.pbs
gen_cced_weather.pbs
gen_ccedp_weather.pbs

Actions

The /cce and /dcce CCE URL actions will show the main CCE listing of CCE objects, or will show other templates if you use other options.

Action options

Category	Use the given category as prefix instead of the standard general template.
Module	Resolves the template name.
Template	Use the given template as part of the template name of standard template.

Example
The request

```
/cce?module=4&template=winterweathersearch
```

uses a template file named gen_winterweathersearch.pbs

Options

If you have a search form for CCE objects on the same page as the listing, the options set in the form will override the options in the tag (unless you use the locked option). You can also put parameters in the URL, which will override the form.

It is possible to display every tag parameter in any CCE object file using the following syntax:

```
<%TagParamXXXX%> where XXXX is the name of a CCE tag parameter
```

you can also list all tag parameters using the following variable:

```
<%AllTagParameters%> Returns all tag parameter with the corresponding values
```

Cached	Disables the cache in special circumstances. By default, this option is not used.
	`Cached=0`
	turns off the caching. This is not recommended. Example:
	`<pbs:cce module=4 class=41 cached=0>`

CacheTime	The option cachetime is used to determine how long a CCE tag should be cached in minutes. Default is 60. Will be overruled by the ini setting [CCE] ForceCacheTime. `<pbs:cce cachetime="10">`

Class	Determines the class of CCE objects that the tags will show. The classes are predefined and match the different types of objects you have in a component. You can setup the classes in the user interface. Example: There is a Weather Data module with ID 4 and the short name *weather*, and there is a Class 41, which belongs to module 4:
	
	`<pbs:cce module=4 class=41>`
	
	SAXOTECH recommends using the class parameter. When you use the <pbs:cce> tag WITHOUT a **class=** parameter, you get a list of objects returned whose class=-1 in the <%dlink%> field of the cce_moduleshortnameitem.pbo file. This will NOT populate the dcce template with the objects field information from the database. In order to build a legitimate dlink from a list of CCE objects for which class is NOT set in the tag, perform a regex operation similar to the following:
	
	`<a href="<%dlink$r("(&Class\x3D-1)=")%>& class=<%class%>">click here</a>`

Component	Replaces the *Module* option. Sets the component used by the tag. The module's short name is used to look up the templates. This option is required. Example: There is a Weather Data module with ID 4 and the short name *weather*. `<pbs:cce module=4>`
Count	Specifies the number of objects per page (Default: 10). Example: `<pbs:cce module=4 class=41 count=15>`
CreatedBy	Enables filtering on creator. **#CurrentUser#** is available for showing the current user's submitted objects.
Datefields	Determines which of the fields should be treated as dates. This option causes the data to be formatted according to the **Dateformat** options. Example: `<pbs:cce module=5 class=51 fields="Birthday" DateFields="Birthday">`
Dateformat	Sets the date format for the date fields. The date fields can also be formatted in the object files.

Dump	Dump all params to a log file. `Dump=1.`

DateRange	Sets both start and end date with the same option. The dates are specified in ISO format:
	`Daterange=20030715-20030722`
	You can set fromdate and todate to the same date by specifying only one date in the daterange field like this:
	`DateRange=20030427`
	Example:
	`<pbs:cce module=4 class=41 daterange="20051101-20051201">`

Days	This option has two uses. It can be used with the **Random** option to determine from how many days the system should show random CCE objects. The second use is to specify how many days back the system should look for CCE objects. Default is 7. When you use the option this way, you must format it like this:
	`Days=7 Fromdate=-1 Todate=YYYYMMDD`
	Fromdate must be set to -1. **Todate** specifies the end date for the list. Example:
	`<pbs:cce module=4 class=41 days=14>`
	Note: If publicus.ini setting CCEFunctionalityMerge=1 in Section CCE is activated, fromdate="-1" need to be removed from the tag.

Defaultdate	Sets **Fromdate** to the last published date. The option is set like this:
	`Defaultdate=1`
	Example:
	`<pbs:cce module=4 class=41 defaultdate=1 todate="20051201">`

Descending	This option is obsolete. Use **SortOrder** instead (see below). Enter
	`Descending=0`
	to use the normal sort order. Enter
	`Descending=1`
	to set the reverse sort order. Example:
	`<pbs:cce module=4 class=41 fields="Temperature" Sortorder="Temperature" Descending=1>`

Execute	Deactivates the initial search. By default, the search is executed. If you enter
	`Execute=0`
	you must set execute to 1 in the search form using a hidden field. Otherwise you will not see any search results. Example:
	`<pbs:cce module=4 class=41 execute=0>`

Fields	Determines which extra custom fields will be available. All fields in the field definitions can be used. The option is set like this: `<pbs:cce module=4 class=41 fields="Temperature,Condition,Wind">`

FIELDMultiSearch	Searches a specified field for multiple keywords. Replace FIELD with the name of the field. The option is used like this:
	`FIELDMultiSearch="text1,text2"`
	Example:
	`<pbs:cce module=4 class=41 fields="Town" TownMultiSearch="Boston,Houston">`

FIELDMultiSearchAnd	Determines that the multiple keyword search in a specified field use AND as the logical operator. The default is to use OR. Replace FIELD with the name of the field. The option is used like this:
	`FIELDMultiSearchAnd=1`
	Example:
	`<pbs:cce module=4 class=41 fields="Town" TownMultiSearch="Las,Vegas" TownMultiSearchAnd=1>`

FIELDSearch	Searches a specified field. Replace FIELD with the name of the field. The option is used like this:
	`FIELDSearch="text"`
	Example:
	`<pbs:cce module=4 class=41 fields="Condition" ConditionSearch="Sunny">`

FIELDSearchExact	Turns off wildcard searches. The text in the criteria must be an exact match. The option is used like this:
	`FIELDSearchExact=1`
	Example:
	`<pbs:cce module=4 class=41 fields="Condition" ConditionSearch="Sunny" ConditionSearchExact=1>`

FromDate	Sets the start date for the list. The date is specified in ISO format: `FromDate=20030715`

FromFIELDSearch **From version 7.1.0sr11**	Searches a specified range using the values specified in FromFIELDSearch and ToFIELDSearch. Only custom fields of type int or float can be used for range searches. If no ToFieldSearch value is specified all fields with values greater than FromFIELDSearch will be returned. Note that when searching in fields of type int a whole number must be used. In fields of type float decimals are allowed. The fields used for range searches must be specified in the Fields parameter on the cce tag. `<pbs:cce FromTemperatureSearch="1.1" ToTemperatureSearch="8.4"FromPopulationSearch="2000" ToPopulationSearch="10000">`
ToFIELDSearch **From version 7.1.0sr11**	Searches a specified range using the values specified in FromFIELDSearch and ToFIELDSearch. Only custom fields of type int or float can be used for range searches. If no FromFieldSearch value is specified all fields with values lower than ToFIELDSearch will be returned. Note that when searching in fields of type int a whole number must be used. In fields of type float decimals are allowed. The fields used for range searches must be specified in the Fields parameter on the cce tag `<pbs:cce FromTemperatureSearch="1.1" ToTemperatureSearch="8.4"FromPopulationSearch="2000" ToPopulationSearch="10000">`
Groups	Sets the group header (not to be confused with group mappings in the ini file). The groups option can only be used on custom fields. For example, if you determine that the field *County* is the group header, the file **ModuleShortNameGroupHeader.pbo** will be output whenever the county changes. The option is set like this: `Groups="County"` The field specified in the groups option will be available as a field in the object file. Example: `<pbs:cce module=4 class=41 fields="Temperature" Groups="Country">` Another example: `Groups="CategoryName"` and the pbo file: `<!-- CCE_townGroupHeader.pbo--> <h2><%CategoryName%></h2>`

IgnoreProfileInUrl	This makes the tag ignore the profile in the URL. `<pbs:cce profile="1453" count="15" module="5" class="256" IgnoreProfileInUrl="1">`
Intfields	Determines which fields should be treated as numbers. This option is deprecated. Use a custom field with the type *Integer* instead.
Locked	This option can only be used in the tag. It locks the tag so it cannot be overridden, either in the form or the URL. This option is useful when you have a sidebar with objects that should not be affected by other CCE tags on the page. The option is set like this: `Locked=1.`
Multisite	If you cooperate with other media sites, you can use this option to list CCE objects from several sites. Example: There is a Weather Data module with ID 4 and the short name *weather*. `<pbs:cce multisite="BA,TY,FC" module=4>`

NavigatorList	It is now possible to add navigators to a Solr search result. This is done by adding the fields to build navigators for to the NavigatorList parameter of the CCE tag that should output the result. The navigators are built using the two templates cce_ModuleShortNameNavigator.pbo and cce_ModuleShortNameNavigatorItem.pbo. Navigators can only be used together with free text search (Solr) or if EmptyCCEListSearchInSolr=1 is used in the SearchEngine section in publicus.ini. The following list of navigators are available:
	`Module, Class, Type, Types, Subtypes, Detailtypes, Date, TaxonomyLeafPath, TaxonomyLeaf, TaxonomyLeafId, Taxonomy, TaxonomyId, CustomField1`
	TaxonomyLeafPath requires version 7.2.0sr20 and Solr schema 7.2.0sr20
	`<pbs:cce navigatorlist="module,class,type,types,subtypes,detailtypes,date,CustomField1" module=4>`

Objectclass	Sets the object class for the object files. The object class is used to look up the object template files for the tag. Example: `<pbs:cce objectclass=49 module=4>` will use object file CCE_Wweather49.pbo if it exists. Otherwise, it will use object file **Weather.pbo**
Profile	Looks up the profile with the given ID and uses the taxonomy expression for that profile to create a query that returns the CCE objects matching the taxonomy expression. If you have a story about a subject that belongs to a profile, you can also show CCE objects that match that profile. Example: `<pbs:!cce module=4 class=41 profile=1334 fromdate="20120301" todate="20120307">` **Note, that in order for profile to work it must be used together with fromdate and todate.**

Published	Used with the **Range** field to control which objects are shown. The option is set like this:
	
	`Published=1`
	
	When you use this option, only objects that have created time within the specified daterange and expires time after ToDate will be shown. If you do not use this option, all objects that have been active within the daterange will be shown. Example:
	
	`<pbs:cce module=4 class=41 daterange="20051101-20051201" published=1 range=1>`

Random	Makes the tag show a random sampling of objects. The option is set like this:
	
	`Random=1`
	
	By default, the tag looks back seven days. You can override this using the **Days** option. Example:
	
	`<pbs:cce module=4 class=41 random=1>`

Range	If you enter the option `Range=1` the system will look for objects that were active within the Daterange. Example:
	
	`<pbs:cce module=4 class=41 daterange="20051101-20051201" range=1>`
	
	* **Rule 1:** Fromdate and Todate are the same and
	
	`Range=1`
	
	All objects that were active on the specified date are shown.
	
	• **Rule 2:**Fromdate and Todate are different and
	
	`Range=1`
	
	All objects that were active within the specified interval are shown.
	• *Rule 3: *Fromdate and Todate are the same and
	
	`Range=0`
	
	All objects that were published on the specified date are shown.
	• **Rule 4:**Fromdate and Todate are different and
	
	`Range=0`
	
	All objects that were published within the specified interval are shown.

RelatedFieldName	Defines which value from the related information table will be shown. By default, this option is not used. Enter
	
	`RelatedFieldName="NAME"`
	
	to make the value of the name field available in the ad. Only one field name can be defined. Example:
	
	`<pbs:cce module=4 class=41 fields="Temperature,Town" RelatedTo="Town" RelatedFieldname="Name">`

Related	Uses a field to look up another field from the related information table. The field you use must be defined in the **Fields** option. By default, this option is not used.Example: If you enter
	`Related="Town"`
	in the tag, information from the town field on the ad will be used to look up information from the related information table.
	`<pbs:cce module=4 class=41 fields="Temperature,Town" Related="Town" RelatedFieldname="Name">`

Search	Freetext search `<pbs:cce module=4 class="41" search="<%text%>">`

SearchTaxonomy	Uses the taxonomy expression for the profile to create a query that will return the CCE objects that match the expression. Syntax for the searches:
	• **Or expression**
	({id1} or {id2} or {id3})
	• **And expression**
	({id1} or {id2} or {id3}) and ({id4} or {id5}) and ({id6} or {id7} or {id8})
	• **And expression with not**
	({id1} or {id2} or {id3}) and not ({id4} or {id5}) and ({id6} or {id7} or {id8})
	Example:
	`<pbs:!cce module=4 class=41 searchtaxonomy="(11 or 12) and not (21 or 22 or 23)">`
	Example of search expressions:
	`19`
	`27 or 34`
	`(11 or 12) and (21 or 22)`
	`(15 or 12) and not (25 or 22)`
	`not (17 or 12) and (27 or 22)`
	`(11 or 12 or 13 or 14) and (21 or 22 or 23) and (31 or 32 or 33 or 34)`
	`(11 or 12 or 13 or 14) and (21 or 22 or 23) and not (31 or 32 or 33 or 34)`

Single	Determines that a single date is sent to the system. This option operates in conjunction with the search form. Please see CCE search form. If this option is not set, you must have from and to date fields. Enter
	
	`Single=1`
	
	to supply a single date, and the system will show the objects for that date. Example:
	
	`<pbs:cce module=4 class=41 single=1>`

SortOrder	Default fields for sorting are: Date, (Category or TypeName), Title, Text and all custom fields.
	From version 7.2.0sr23. it's also possible to sort by PlainPriority and ProfilePriority.
	
	You can for example sort by a custom field named Temperature like this:
	
	`<pbs:cce module=4 class=41 fields="Temperature" Sortorder="Temperature">`
	
	SortOrder will sort your data ascending by default. To reverse the order, use
	
	`Descending=1`
	
	(see above), or enter
	
	`(DESC)`
	
	after the field by which you want to sort. You can mix ascending and descending sort orders:
	
	`<pbs:cce ... Groups="FieldName1" SortOrder="FieldName2(ASC),FieldName3(DESC),FieldName4">`
	
	Sort by date
	
	`<pbs:cce ... Sortorder="date" Descending=1>`
	
	When sorting a list of CCE Events by date, use EventDateSort=1 instead of the SortOrder option.
	
	From version 7.2.0. it's also possible to sort by EventDate, EventFromTime and EventToTime. This will only work with ShowEventOccurrences enabled.
	
	`<pbs:cce SortOrder="EventDate(asc),EventFromTime(asc),Title(desc)" ShowEventOccurrences=1>`

Start	Sets the start record. This option is typically not used manually. It is used automatically by the system when you have multiple pages. Example: `<pbs:cce module=4 class=41 start=16 count=15>` SortOrder="EventDate(asc),EventFromTime(asc),Title(desc)" ShowEventOccurrences=1
TaxonomyWords	Searches for CCE objects that have the taxonomy words given in the list. It should be a list of IDs for taxonomy words: `TaxonomyWords="23,45,56"`
TaxonomyWordsAnd	This option is set to 1 if the objects should have all the taxonomy words given in the list: `TaxonomyWordsAnd=1`
ToDate	Sets the end date for the list. The date is specified in ISO format: `ToDate=20030715`
Type	Specifies the CCE type (category). This option is only used when listing one object, which belongs to a single type.Example:* * `<pbs:cce module=4 class=41 Type="CAT2901" >`
Types	Works like **Type**, but you can specify multiple types (categories). The types must be specified in a comma separated list.Example: `<pbs:cce module=4 class=41 Types="CAT2901,CAT2950">`

Reset drop-down field values

You can use a JavaScript to reset the drop-down fields to the same values that they had before the form was submitted. The following JavaScripts set the value for the fields ToDate, FromDate and Type:

⚠ JavaScript is case sensitive. You must write the field names exactly as they are in the form. This script must be placed at the bottom of the gen_cce_ModuleShortName.pbs template.

```
<script language="javascript">
var a = document.classfsearch.ToDate.options;
for (x=0;x<a.length;x++) {
if (("<pbs:!#ToDate#>" == "") && (a[x].value == "-1")) {a[x].selected = true;}
else
if (a[x].value == "<pbs:!#ToDate#>") {a[x].selected = true;}
}
a = document.classfsearch.FromDate.options;
for (x=0;x<a.length;x++) {
if (("<pbs:!#FromDate#>" == "") && (a[x].value == "-1")) {a[x].selected = true;}
else
if (a[x].value == "<pbs:!#FromDate#>") {a[x].selected = true;}
}
a = document.classfsearch.Type.options;
for (x=0;x<a.length;x++) {
if (a[x].value == "<pbs:!#Type#>") {a[x].selected = true;}
}
</script>
```

Objects with modules of module type Event

There are parameters available that can filter on the special properties for Event objects:

EventFromDate	At least one event occurs on or after the given date (yyyymmdd format) (default none)
EventToDate	At least one event occurs on or before the given date (yyyymmdd format) (default none)
EventDateSort	If set to 1 the search result will be sorted by event date (default 0). It cannot be combined with other sorting parameters.
ShowEventOccurrences	If set to 0 objects appears once, on the first following event occurrence. If set to 1, the object appears once for every event occurrence. (default 0)

- CCE_ModuleShortName.pbo
- CCE_ModuleShortNameGroupHeader.pbo
- CCE_ModuleShortNameItem.pbo
- CCE_ModuleShortNameItemAccessControls.pbo
- CCE_ModuleShortNameItemAccessControlsItem.pbo
- CCE_ModuleShortNameItemEventDateItem.pbo
- CCE_ModuleShortNameItemTaxonomy.pbo
- CCE_ModuleShortNameItemTaxonomyItem.pbo
- CCE_ModuleShortNameItemTaxonomyItemPathItem.pbo
- CCE_ModuleShortNameNavigator.pbo
- CCE_ModuleShortNameNavigatorFilter.pbo
- CCE_ModuleShortNameNavigatorFilterItem.pbo
- CCE_ModuleShortNameNavigatorItem.pbo
- CCE_ModuleShortNamePageLink
- CCE_ModuleShortNamePageLinkItem
- CCE_MPLACE.pbo
- CCE_MPLACEitem.pbo
- CCE-search_ovr.pbs

CCE_ModuleShortName.pbo

Contains the framework that is output once. This object file is required.

Example file name

cce_weather.pbo

Example object file

```
<html>
<body>
<table><%items%></table>
<a href="<%SortByTemperature%>">Sort By Temperature</a>
</body>
</html>
```

<%FirstHit%>	The number for the first record being viewed.
<%Items%>	Outputs the CCE items.
<%LastHit%>	The number for the last record being viewed.
<%NextLink%>	Contains a link to the next page, when not on the last page. It must be posted to the server with JavaScript like this: `<a href="javascript:dosubmit('<%NextLink%>');"> \\ Next »</a>`
<%PrevLink%>	Contains a link to the previous page, when not on the first page. It must be posted to the server with JavaScript like this: `<a href="javascript:dosubmit('<%PrevLink%>');"> « Previous</a>`
<%SortByTitle%>	Contains a link that will sort all the objects by title. It must be posted to the server with JavaScript like this: `<a href="javascript:dosubmit('<%SortByTitle%>');">Published</a>` The system will remember the sort order so you can reverse the list. To keep to the sort order when you browse using a search form, you must add a field to the form that remembers the value.
<%SortByDate%>	Contains a link that will sort all the objects by date. It must be posted to the server with JavaScript like this: `<a href="javascript:dosubmit('<%SortByDate%>');">Published</a>` The system will remember the sort order so you can reverse the list. To keep to the sort order when you browse using a search form, you must add a field to the form that remembers the value.

<%SortBy....%>	You can also use custom fields for the sorting in addition to date and title. Replace "..." with the name of the field. Example: `<%SortByTemperature%>`
<%TotalHits%>	Shows the total number of records for the category being viewed.
<%SearchNavigator%> **<%search_navigator%>**	The navigators build when doing search. This is only available when the **NavigatorList** parameter is given. The navigator is rendered through the CCE_ModuleShortNameNavigator.pbo and CCE_ModuleShortNameNavigatorItem.pbo. **In order to show a more readable syntax the setting UseNavigatorFilters=1 is needed in section SearchEngine in publicus.ini. This will also change the format of the URI returned by the navigators to be more readable. Since this is a sytem wide setting it will also influence navigators in story search.**
<%SearchNavigatorFilters%>	This is only available when the **NavigatorList** parameter is given. The filters that are selected in the navigators can be shown using CCE_ModuleShortNameNavigatorFilter.pbo and CCE_ModuleShortNameNavigatorFilterItem.pbo. **In order to show the filters the setting UseNavigatorFilters=1 is needed in section SearchEngine in publicus.ini. This will also change the format of URI returned by the navigators to be more readable.**
<%PageNo%>	Contains which pagenumber in the search result you are on (added from 6.3.0sr12)
<%PageCount%>	Contains the nunber of pages the search result consists of (added from 6.3.0sr12)
<%PageSize%>	Contains the size of a page (same as count on pbs:cce-tag) (added from 6.3.0sr12)
<%PageLinks%>	Outputs links to all pages for the curren search result. See templates CCE_ModuleShortnamePageLink.pbo and CCE_ModuleShortnamePageLinkItem.pbo.

CCE_ModuleShortNameGroupHeader.pbo

Contains a single field with the same name as the field specified by the **Groups** option. If you use the option

```
Groups="County"
```

you will have the field

```
<%county%>
```

in the object file. If you have specified a date field in the **Groups** option, the field will be formatted as a date if it is also listed in the **Datefields** option.

This object file is required.

Example file name

cce_weatherGroupHeader.pbo

Example object file

```
<tr>
<td>
<b>=========== <%Temperature%> ===============</b>
</td>
</tr>
```

CCE_ModuleShortNameItem.pbo

This file is output for each CCE object. This object file is required.

Example file name

CCE_WeatherItem.pbo

Example object file

```
<tr>
  <td>
    Town: <%Town%> - <%Date%><br>
    Condition: <%Condition%><br>
    Temperature: <%Temperature%>
  </td>
</tr>
```

<%AvgRanking%>	Contains the average ranking of a CCE object.
<%BayesianRanking%>	Bayesian ranking of a CCE object.
<%Class%>	Outputs the **Class id** parameter from the tag.
<%ClassName%>	Outputs the **Class name** parameter from the tag.

<%Count%>	This field starts with the value 1 and is increased by one for each object.
	`[%{count=1}Object 1%]`
	`View add 1`
	`[%{count=2|count=3}Object 2 and 3%]`

<%Date%>	The date for the CCE object formatted based on DateFormat paramter of not defined then use the default on the server.
<%Dlink%>	A link to the detail page for the ad.
<%Expires%>	The date the ad expires.
<%FieldName%>	Outputs the number value of the extra custom field with **FieldName**.
<%GUID%>	Outputs the item's GUID.
<%Id%>	The CCE object ID.
<%IsLastItem%>	Will be '1' for the last item, else blank.

<%ISOCreatedDateTime%> **From version 7.2.0sr22**	The creation date and time in ISO format. (YYYYMMDDHHMMSS). This variable should be used in conjunction with date formatting functions. Please read the section Formatting dates. This field can be formatted using object script. `<%ISOCreatedDateTime%>`
<%ISOModifiedDateTime%> **From version 7.2.0sr22**	The last modified date and time in ISO format. (YYYYMMDDHHMMSS). This variable should be used in conjunction with date formatting functions. Please read the section Formatting dates. This field can be formatted using object script.
<%ImageLink%>	A link to the image, if the ad is a display ad. If multiple images are attached to the CCE object, use a regular expression to replace the 'element' query parameter and indicate which image to display (&element=[1, 2,... n]). It must be used like this: `<img src="<%imagelink$r("(Element\x3d)1=Element=2")%>&maxw=100" />`
<%LogoLink%>	A link to the logo, if the ad is a display ad. It must be used like this: `<img src="<%LogoLink%>&MaxW=320" border=0>`
<%ModuleType%> **From version 7.2.0**	The module type of the CCE object (normal, event, asset)
<%NumImages%>	Outputs the number of images attached to the CCE Object. The logo image is not counted in this variable.
<%ObjectDate%>	The date for the CCE object in YYYYMMDD format.
<%OrgCCEID%>	Contains the original ID of the object. Required for allowing the user to update the object. It only has a value if the user is the creator of the object and therefore can edit it.
<%PrintLink%>	A link to the printer friendly page. It must be used like this: `<a href="<%PrintLink%>">Print</a>`
<%RankingLink%>	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL '<%RankingLink%>' + '&rating=' + y Where y is the ranking given by a user.
<%Relatedinfo%>	Outputs the information from the related field when you use the **Relatedto** function.
<%TaxonomyList%> **From version 6.9.0sr10**	Contains the taxonomy words of the CCE object. See CCE_ModuleShortNameItemTaxonomy.pbo
<%Text%>	Outputs the defined field **Text**, which is defined in the ini file.

<%TipLink%>	A link to the tip a friend page. It must be used like this: `<a href="<%TipLink%>">Tip a friend</a>`
<%Title%>	The title field for the ad.
<%TotalCount%>	Contains the total number of items returned.
<%TotalNoofVotes%>	Contains the total number of votes on the CCE object.
<%UserSubmitDeleteLink%>	If the user is the creator of the item, this variable is available. It contains a link that if visited by the creator will delete the displayed object.

Type information

The variables in this section shows the types, sub types and detail types for a CCE object. All variables prefixed Category are duplicates of variables prefixed Type.

Some of these require version 6.9.0sr29 or higher.

<%Type%>	Contains the first type for the object.
<%Type01...09%>	The additional types for the object.You can use `<%Type01%>, <%Type02%>` etc. for additional categories.
<%TypeName%>	The first type name for the object.
<%TypeName01...09%>	The additional type names.
<%SubType%>	The first subtype for the object.
<%SubType01...09%>	The additional subtypes for the object.
<%SubTypeName%>	The first subtype name for the object.
<%SubTypeName01...09%>	The additional subtype names.
<%DetailType%>	The first detail type for the object.
<%DetailType01...09%>	The additional detail types for the object.
<%DetailTypeName%>	The first detail type name for the object.
<%DetailTypeName01...09%>	The additional detail type names.
<%Category%>	This field can be used instead of **<%Type%>**
<%Category01...09%>	The **<%Category01...09%>** fields can be used instead of **<%Type01...09%>**
<%CategoryName%>	This field can be used instead of **<%Typename%>**.
<%CategoryName01...09%>	Can be used instead of **<%Typename%>** fields.
<%SubCategory%>	The first Subtype for the object.
<%SubCategory01...09%>	The additional subtypes for the object.
<%SubCategoryName%>	The first Subtype name for the object.
<%SubCategoryName01...09%>	The additional subtype names.
<%DetailCategory%>	The first detail type for the object.

<%DetailCategory01...09%>	The additional detail types for the object.
<%DetailCategoryName%>	The first detail type name for the object.
<%DetailCategoryName01...09%>	The additional detail type names.

Displaying Event objects

The variables in this section are only available if the object belongs to an Event module and the <pbs:cce> parameters **EventFromDate** and **EventToDate** are used.

<%EventDate%>	Contains the first coming occurrence of the object. If ShowEventOccurrences=1 is on the pbs:cce tag, <%EventDate%> will output the current occurrence as defined on the EventFromDate and EventToDate parameters.
<%EventDateList%>	Contains a list of the first many object occurrence dates. The dates are formatted as yyyymmdd,yyyymmdd....
<%EventDateListSeparated%>	Contains a list of first many object occurrence dates and times. The variable uses object file CCE_ModuleShortNameItemEventDateItem.pbo which contains <%EventDate%> and similar variables. This means that users can now format the date using $d modifier (<%EventDate$d("yyyy/mm/dd")%>)
<%ShowEventOccurrences%>	Contains the value of the <pbs:cce> tag parameter of **ShowEventOccurrences**

The following variables describe the recurrence pattern attached to the object and are only available if the object belongs to an Event module and the <pbs:cce> parameters **EventFromDate** and **EventToDate** are used.

<%EventFromDate%>	Contains the date from which the system is to calculate occurrences (yyyymmdd format)
<%EventToDate%>	Contains the date to which the system is to calculate occurrences (yyyymmdd format) - it is used when **Stop on** is set in Edit UI.
<%NumberOfOcc%>	Contains how many occurrences to calculate from the **EventFromDate** - it is used when **Stop after** is set in Edit UI.
<%StopMethod%>	Contains how the system is to stop calculating occurrences. 0 is on a set date, 1 is after a certain number of occurrences.
<%EventFromTime%>	Contains from what time of day the object occurs. The format of this variable has changed to 00000000HHMM which means that users can now use $d modifier to format the value this variable (<%EventFromTime$d("hh:mm AM/PM")%>).
<%EventToTime%>	Contains to what time of day the object occurs. The format of this variable has changed to 00000000HHMM which means that users can now use $d modifier to format the value this variable (<%EventFromTime$d("hh:mm AM/PM")%>).
<%Pattern%>	Contains the overall method used to calculate occurrences.
<%Subpattern%>	Contains the specific method used to calculate occurrences.
<%Jump1%>	Contains the first specific value in the pattern.
<%Jump2%>	Contains the second specific value in the pattern.
<%ActiveDays%>	Contains what days in the week are selected in the pattern.
<%Day1%>	If Sunday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day2%>	If Monday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day3%>	If Tuesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.

<%Day4%>	If Wednesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day5%>	If Thurday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day6%>	If Friday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day7%>	If Saturday is in the **ActiveDays**, then it has value 1, otherwise it has no value.

The way a pattern is used to calculate object occurrences is described in UserSubmitFormCCERecurrencePattern.pbo.

Access Control descriptors

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of CCE_Module ShortNameItemAccessControls.pbo
<%CurrentFlowItemId%> **From version 7.1.0**	Represents the current access element in the flow as an ID number. This variable is used in conjunction with Access Control.
<%CurrentFlowItemIdent%> **From version 7.1.0**	Represents the current access element in the flow as the identifier. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%> **From version 7.1.0**	Outputs the standard text field of the current access element in the flow. This variable is used in conjunction with Access Control.
<%SingleSaleId%> **From version 7.1.0**	This field contains the ID of the access control element attached to the item.
<%SingleSaleIdent%> **From version 7.1.0**	This field contains the identifier of the access control element attached to the item.
<%SingleSaleText%> **From version 7.1.0**	This field contains the text of the access control element attached to the item.

Displaying Asset objects From version 7.2.0

The variables in this section are only available if the object belongs to an Asset module.

<%External%>	Shows if the asset is external. Values (0,1)
<%AssetTypeID%>	ID of the assettype connected to the CCE item
<%AssetTypeName%>	Name of the assettype connected to the CCE item
<%VirtualFolderID%>	ID of virtual folder used by the asset
<%VirtualFolderName%>	Name of the virtual folder.
<%VirtualFolderFilePath%>	FilePath for the virtual folder
<%VirtualFolderStatus%>	Status of the virtual folder. (protected, public)
<%FileSize%>	Size of the asset file in bytes
<%FileName%>	Name of the asset file
<%AssetURL%>	URL to the binary file

CCE_ModuleShortNameItemAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of CCE_Module ShortNameItemAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

CCE_ModuleShortNameItemAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

CCE_ModuleShortNameItemEventDateItem.pbo

Outputs the CCE event objects' occurrences.

<%EventDate%>	Outputs the occurrence's event date, can be formatted using $d. `<%EventDate$d("yyyy/mm/dd")%>`
<%EventFromTime%>	Outputs the occurrence's from time. `<%EventFromTime$d("hh:nn")%>`
<%EventToTime%>	Outputs the occurrence's to time, if one is specified. `<%EventToTime$d("hh:nn")%>`

⚠ Will work with Solr from version 6.8.0sr17

CCE_ModuleShortNameItemTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

CCE_ModuleShortNameItemTaxonomyItem.pbo

<%ID%>	Outputs the ID of the taxonomy word.
<%Description%>	Outputs the taxonomy word text.
<%Path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.

<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

CCE_ModuleShortNameItemTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

CCE_ModuleShortNameNavigator.pbo

This is the main file for outputting cce navigators.

Example file name

cce_SMSNameNavigator.pbo

Example object file

```
[%{count>1}
<td>
<form name="ccefsearch" method="post" action="/apps/pbcs.dll/cce?module=2">
<%NavigatorName%><br>
<select type="text" name="search" style="width:240px;margin:5px 0 5px 0;">
<%items%>
</select>
<input type=submit name=Go VALUE="Submit">
</form>
</td>
%]
```

Variables

<%NavigatorName%>	The name of the facet being build.
<%Items%>	Outputs the Facet elements.
<%Count%>	The number of facet elements in items.

CCE_ModuleShortNameNavigatorFilter.pbo

Holds all the selected navigator filters.

<%Items%>	All the filter items.

CCE_ModuleShortNameNavigatorFilterItem.pbo

Used to generate each individual selected filter.

<%FilterName%>	The name of the filter item.

<%FilterValue%>	Value used in selected filter.
<%RemoveFilterUrl%>	Remove the selected filter from the search result.

CCE_ModuleShortNameNavigatorItem.pbo

This is used for outputting each element in the navigator list.

Example file name

cce_SMSNameNavigatorItem.pbo

Example object file

```
<option value="<%ModifierValue%>" ><%Counter%> - <%ModifierName%> 
(<%ModifierHits%>)</option>
```

Variables

<%ModifierHits%>	The number of hits on this element in the navigator list.
<%ModifierName%>	The name of the element in the navigator list.
<%ModifierValue%>	The encrypted search string. Should be assigned to a post or get variable named search.
<%Counter%>	The number of the current item.

CCE_ModuleShortNamePageLink

Example file name

cce_GOLFPageLink.pbo

Example object file

```
Here we have <%count%> page links: <%items%>
```

Variables

<%Items%>	Outputs the pagelink-items elements.
<%Count%>	Outputs the number of pagelinks

Note that this functionality is first available from 6.3.0sr12.

CCE_ModuleShortNamePageLinkItem

This is the main file for outputting a pagelinkitem.

Example file name

cce_GolfPageLinkItem.pbo

Example object file

```
<a href="<%PageLink%>">Page <%LinkPageNo%> of <%PageCount%>, which is items
<%FirstHit%> through <%LastHit%>. Current page has no <%PageNo%></a><br />
```

Variables

<%PageNo%>	The page number currently being shown.
<%PageCount%>	The number of pages in total
<%PageLink%>	The link to another result page
<%LinkPageNo%>	The page number for the above link
<%Counter%>	Same as <%LinkPageNo%>
<%FirstHit%>	Object number of the first item in result list for the link
<%LastHit%>	Object number of the last item in result list foir the link

Note that this functionality is first available from 6.3.0sr12.

CCE_MPLACE.pbo

CCE_MPLACEitem.pbo

CCE-search_ovr.pbs

pbs ccedetails

<pbs:ccedetails>

The CCE detail page is used to show all details about a CCE object. This page is typically linked to from the listing of CCE objects.

> ⚠ It is possible to only specify an ID or GUID when retrieving a CCE object. This works as both URL parameters on actions and on the details tag.

The tag is used like this:

```
<pbs:!ccedetails module=11 datefields="Date" dateformat="mmm dd, yyyy">
```

Or like this:

```
<pbs:!ccedetails module=4 class=41 date="20051101" type="CAT1001" id="23533221" >
```

Template

The template file for CCE object details is called **gen_cced_ModuleShortName.pbs**.

Example: **gen_cced_weather.pbs**.

The template for the printer-friendly version is called **gen_cced_ModuleShortNamep.pbs**. The printer friendly version is made for users that would like to print a CCE object. The template should contain very few graphic elements and design.

Options

Cachetime	The CCE object's cachetime.
Class	Determines which class of CCE objects the tags will show.
Date	The CCE object's publication date.
Datefields	Determines which of the fields in the database contain the date information.

Dateformat	Sets the date format, which is to be used when the system displays the objects.
Fields	A list of extra custom fields that should be shown.
ID	The CCE object's ID.
Module	Sets the module used by the tag. The component's short name is used to look up the templates. This option is required.Example: There is a *Weather Data* module with ID 4 and the short name *weather:* `<pbs:ccedetails module=4 class=41 date="20051101" type="CAT1001" id="23533221">`
Related	Pulls data from the related table in the database. This is typically customer information. If the related field is called CUSTOMER, you can use this option: `Related=CUSTOMER` If you use this option, the object file **CCE_ShortModuleNameRelated.pbo** is used to show the related information.
Objectclass	Controls which object file will be used. Enter `Objectclass="1"` to make the system look for CCE_ModuleShortNameDetails1.pbo
Type	The CCE object's type.

CCE_ModuleShortNameDetails.pbo

<%AdClass%>	Output Class ID
<%AttachedLink%>	Output AttachedLink
<%AvgRanking%>	Contains the average ranking of a CCE object.
<%CCEType%>	Output CCE Type
<%Class%>	Output Class ID
<%ClassName%>	Outputs the **Class name** parameter from the tag.
<%Date%>	Created date like: 20100622
<%Expires%>	The date the object expires.
<%GUID%>	Outputs the item's GUID.
<%ID%>	Output Item ID
<%ImageLink%>	A link to the first image attached to the CCE object. If multiple images are attached to the CCE object, use a regular expression to replace the 'element' query parameter and indicate which image to display (&element=1, 2,... n). `<img src="<%imagelink$r("(Element\x3d)1=Element=2")%>&maxw=100" />`

<%ISOCreatedDateTime%> **From version 7.2.0sr22**	The creation date and time in ISO format. (YYYYMMDDHHMMSS). This variable should be used in conjunction with date formatting functions. Please read the section Formatting dates. This field can be formatted using object script. `<%ISOCreatedDateTime%>`
<%ISOModifiedDateTime%> **From version 7.2.0sr22**	The last modified date and time in ISO format. (YYYYMMDDHHMMSS). This variable should be used in conjunction with date formatting functions. Please read the section Formatting dates. This field can be formatted using object script.
<%LogoLink%>	URL for logo link
<%Longitude%>	Outputs the longitude for the object (first available in 6.5.0 version line)
<%Module%>	Output Module ID
<%NumImages%>	Outputs the number of images for the object.
<%ObjectDate%>	The date for the CCE object in YYYYMMDD format.
<%OrgCCEID%>	Contains the original ID of the object. Required for allowing the user to update the object. It only has a value if the user is the creator of the object and therefore can edit it.
<%PrintLink%>	URL for print version
<%RankingLink%>	The RankingLink does not contain the actual ranking value this must be added to the link by using client side scripting. The Following must be added to the link before submitting the link to the DLL '<%RankingLink%>' + '&rating=' + y Where y is the ranking given by a user.
<%RelatedInfo%>	Pulls in the related information object file.
<%Site%>	Output site code
<%TaxonomyList%> **From version 6.9.0sr10**	Contains the taxonomy words of the CCE object. See CCE_ModuleShortNameDetailsTaxonomy.pbo
<%Text%>	Output the text field
<%TipLink%>	URL for tip a friend version
<%Title%>	Output title
<%TotalNoofVotes%>	Contains the total number of votes on the CCE object.
<%UserCommentsLink%> **From version 6.9.0sr19**	Link to comments on the cce object
<%UserCommentsStatus%> **From version 6.9.0sr19**	Outputs value 1 or 3 . value 1 means that user comments are activated on the CCE object, value 3 means that user comments are locked. If the variable is empty, this means that user comments are not activated
<%UserSubmitDeleteLink%>	If the user is the creator of the item, this variable is available. It contains a link that if visited by the creator will delete the displayed object.

Type information

The variables in this section shows the types, sub types and detail types for a CCE object. All variables prefixed Category are duplicates of variables prefixed Type.

Some of these require version 6.9.0sr29 or higher.

<%Type%>	Contains the first type for the object.
<%Type01...09%>	The additional types for the object.You can use `<%Type01%>, <%Type02%>` etc. for additional categories.
<%TypeName%>	The first type name for the object.
<%TypeName01...09%>	The additional type names.
<%SubType%>	The first subtype for the object.
<%SubType01...09%>	The additional subtypes for the object.
<%SubTypeName%>	The first subtype name for the object.
<%SubTypeName01...09%>	The additional subtype names.
<%DetailType%>	The first detail type for the object.
<%DetailType01...09%>	The additional detail types for the object.
<%DetailTypeName%>	The first detail type name for the object.
<%DetailTypeName01...09%>	The additional detail type names.
<%Category%>	This field can be used instead of **<%Type%>**
<%Category01...09%>	The **<%Category01...09%>** fields can be used instead of **<%Type01...09%>**
<%CategoryName%>	This field can be used instead of **<%Typename%>**.
<%CategoryName01...09%>	Can be used instead of **<%Typename%>** fields.
<%SubCategory%>	The first Subtype for the object.
<%SubCategory01...09%>	The additional subtypes for the object.
<%SubCategoryName%>	The first Subtype name for the object.
<%SubCategoryName01...09%>	The additional subtype names.
<%DetailCategory%>	The first detail type for the object.
<%DetailCategory01...09%>	The additional detail types for the object.
<%DetailCategoryName%>	The first detail type name for the object.
<%DetailCategoryName01...09%>	The additional detail type names.

Displaying Event objects

The variables in this section are only available if the object belongs to an Event module and the <pbs:cce> parameters **EventFromDate** and **EventToDate** are used.

<%EventDate%>	Contains the first coming occurrence of the object.
<%EventDateList%>	Contains a list of the first many object occurrence dates. The dates are formatted as yyyymmdd,yyyymmdd....
<%EventDateListSeparated%>	Contains a list of first many object occurrence dates and times. The variable uses object file CCE_ModuelShortNameItemEventDateItem.pbo which contains <%EventDate%> and other variables. This means that users can now format the date using $d modifier (<%EventDate$d("yyyy/mm/dd")%>)
<%ShowEventOccurrences%>	Contains the value of the <pbs:cce> tag parameter of **ShowEventOccurrences**

The following variables describe the recurrence pattern attached to the object and are only available if the object belongs to an Event module and the <pbs:cce> parameters **EventFromDate** and **EventToDate** are used.

<%EventFromDate%>	Contains the date from which the system is to calculate occurrences (yyyymmdd format)
<%EventToDate%>	Contains the date to which the system is to calculate occurrences (yyyymmdd format) - it is used when **StopMethod** is 0.
<%NumberOfOcc%>	Contains how many occurrences to calculate from the **EventFromDate** - it is used when **StopMethod** is 1.
<%StopMethod%>	Contains how the system is to stop calculating occurrences. 0 is on a set date, 1 is after a certain number of occurrences.
<%EventFromTime%>	Contains from what time of day the object occurs. From version 6.5.0sr26 the format of this variable has changed to 00000000HHMM which means that users can now use $d modifier to format the value this variable (<%EventFromTime$d("hh:mm AM/PM")%>).
<%EventToTime%>	Contains to what time of day the object occurs. From version 6.5.0sr26 the format of this variable has changed to 00000000HHMM which means that users can now use $d modifier to format the value this variable (<%EventFromTime$d("hh:mm AM/PM")%>).
<%Pattern%>	Contains the overall method used to calculate occurrences.
<%Subpattern%>	Contains the specific method used to calculate occurrences.
<%Jump1%>	Contains the first specific value in the pattern.
<%Jump2%>	Contains the second specific value in the pattern.
<%ActiveDays%>	Contains what days in the week are selected in the pattern.
<%Day1%>	If Sunday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day2%>	If Monday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day3%>	If Tuesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day4%>	If Wednesday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day5%>	If Thurday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day6%>	If Friday is in the **ActiveDays**, then it has value 1, otherwise it has no value.
<%Day7%>	If Saturday is in the **ActiveDays**, then it has value 1, otherwise it has no value.

The way a pattern is used to calculate object occurrences is described in UserSubmitFormCCERecurrencePattern.pbo.

Access Control descriptors

<%AccessControls%> **From version 7.3.0sr1**	Outputs the access control items using the object file of CCE_Module ShortNameDetailsAccessControls.pbo
<%CurrentFlowItemId%> **From version 7.1.0**	Represents the current access element in the flow as an ID number. This variable is used in conjunction with Access Control.
<%CurrentFlowItemIdent%> **From version 7.1.0**	Represents the current access element in the flow as the identifier. This variable is used in conjunction with Access Control.
<%CurrentFlowStatusText%> **From version 7.1.0**	Outputs the standard text field of the current access element in the flow. This variable is used in conjunction with Access Control.

<%Protected%> **From version 7.1.0**	This variable is used when the pbs:ccedetails tag is used on a page, which is not processed by the /dcce action, so the template will handle how the access control protected element should look.
<%SingleSaleId%> **From version 7.1.0**	This field contains the ID of the access control element attached to the item.
<%SingleSaleIdent%> **From version 7.1.0**	This field contains the identifier of the access control element attached to the item.
<%SingleSaleText%> **From version 7.1.0**	This field contains the text of the access control element attached to the item.

Displaying Asset objects From version 7.2.0

The variables in this section are only available if the object belongs to an Asset module.

<%External%>	Shows if the asset is external. Values (0,1)
<%AssetTypeID%>	ID of the assettype connected to the CCE item
<%AssetTypeName%>	Name of the assettype connected to the CCE item
<%VirtualFolderID%>	ID of virtual folder used by the asset
<%VirtualFolderName%>	Name of the virtual folder.
<%VirtualFolderFilePath%>	FilePath for the virtual folder
<%VirtualFolderStatus%>	Status of the virtual folder. (protected, public)
<%FileSize%>	Size of the asset file in bytes
<%FileName%>	Name of the asset file
<%AssetURL%>	URL to the binary file

CCE_ModuleShortNameDetailsAccessControls.pbo

<%Items%>	Outputs the access control items using the object file of CCE_Module ShortNameDetailsAccessControlsItem.pbo
<%TotalCount%>	Outputs the number of access control items in the list.

CCE_ModuleShortNameDetailsAccessControlsItem.pbo

<%ID%>	Outputs the ID of the access control item.
<%Identifier%>	Outputs the identifier string of the access control item.
<%Name%>	Outputs the display name of the access control item.
<%MessageText%>	Outputs the standard message text used to describe the access control item.
<%Counter%>	The access control item's number in the list.
<%TotalCount%>	The total number of access control items in the list.
<%IsLast%>	Has the value of 1 if the access control item is the last one in the list.

CCE_ModuleShortNameDetailsEventDateListItem.pbo

Outputs the CCE event objects' occurrences.

<%EventDate%>	Outputs the occurrence's event date, can be formatted using $d. `<%EventDate$d("yyyy/mm/dd")%>`
<%EventFromTime%>	Outputs the occurrence's from time. `<%EventFromTime$d("hh:nn")%>`
<%EventToTime%>	Outputs the occurrence's to time, if one is specified. `<%EventToTime$d("hh:nn")%>`

CCE_ModuleShortNameDetailsTaxonomy.pbo

<%Items%>	Outputs the taxonomy items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the number of taxonomy items in the list.

CCE_ModuleShortNameDetailsTaxonomyItem.pbo

<%id%>	Outputs the ID of the taxonomy word.
<%description%>	Outputs the taxonomy word text.
<%path%>	Outputs the parent tree of the taxonomy word. The delimiter parameter is used to seperate items in the tree.
<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo to output the tree from the root.
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

CCE_ModuleShortNameDetailsTaxonomyItemPathItem.pbo

<%parent%>	Outputs The description of the parent taxonomy word.
<%islast%>	Outputs 1 if the item is the last in the tree.
<%isfirst%>	Outputs 1 if the item is the first in the tree.

CCE_ModuleShortNamePrintDetail.pbo

Used for the printer-friendly version of the CCE object.

The same fields as in CCE_ModuleShortNameDetails.pbo are available.

CCE_ModuleShortNameRelated.pbo

If you use the **Relatedto** option, you can use this field to show the related information. The fields that are available match the fields you have in the related table.

pbs ccefields

<pbs:ccefields>

Used to extract information from the CCE data and metadata. The tag can be used to populate the search form with relevant data.

This tag must use one or more options and can be used with the exclamation mark option.

Object files will only be used when the parameter

```
Optionlist=0
```

is used. FieldName is replaced with the value from the field option.

Options

All	Will return all types, not just ones with data attached. By default, all is not an available option. Enter `All=1` to get all types, not just the ones with data attached.
Any	Turns on *Any* as a choice in a drop-down menu. This function can only be used when you output an **optionlist**. By default, *Any* is not available as an option. Enter `Any=1` to add *Any* as a choice in the drop-down menu.
Class	Sets the class. This field must be set, if it is used. Enter `Class="x"` where x is the class.
Columns	Determines how many columns the option list will have. This option can only be used when `Optionlist=0`. The option is set like this: `Columns=3` When you use this option, you must use the cce_ModuleShortNameFieldNameRow.pbo file

CustomFields	Returns all custom fields that are defined. `All=1` denoting all the fields. Example: `<pbs:!ccefields module=4 class=41 field="CustomFields" optionlist=0 any=0 all=1>`
Date	Makes the system display the dates for the days in the last week. It will also display options for the last 7, 14 and 30 days.
Dateformat	Sets the date format for date fields. Dateformat is set by using the codes found in the section Formatting dates. For example: `Dateformat="mm/dd/yyyy"`
Detailtype	Lists the detail types defined for the class. Example: `<pbs:!ccefields module=4 class=41 field="DetailType" optionlist=1 any=1>`
Excludevalues	Excludes values from the option list that is generated. By default, this option is not used. Enter a comma separated list of values that should be excluded. Example: `<pbs:!ccefields module=4 class=41 field="JOBOPTION" excludevalues="1,2">` This tag builds a option list of the values of the extra field JOBOPTION, but will not include the values 1 and 2.
Field	Determines which field will be shown.
FieldName	Lists the values of a given extra custom field. Example: `<pbs:!ccefields module=4 field="Temperature" optionlist=1 any=1>`
Module	Determines the component from which data will be shown. This option is required.

Objectclass	Sets the objectclass. It is only valid when `Optionlist=0`

Optionlist	Turns off the output of an optionlist, and uses object files instead. The default is
	`Optionlist=1.`
	Enter
	`Optionlist=0`
	to use object files instead of an objectlist.

Related	Specifies the information that will be shown from the related information table. You can then use the related fields with the field option. Example:
	`<pbs:!ccefields module=4 class=41 Related="Town" field="Hotel">`
	NOTE: If this option is used in conjunction with 'optionlist=0', the standard ccefields object files will not be used. Instead, the tag will look for file names of the form cce_[module short name]fields[related value][field value].pbo where [related value] is the value of the option 'related' in the ccefields tag and [field value] is the value of the option 'field' in the tag. For example, assuming module 4 is called 'staff', the following tag will look for files cce_staffFieldsStaff_lastname_initialName.pbo and cce_staffFieldsStaff_lastname_initialNameItem.pbo
	`<pbs:ccefields module=4 Related=staff_lastname_initial field=Name optionlist=0 all=1>`

Subtype (Subcategory)	Lists the subtypes defined for the class. Example: `<pbs:!ccefields module=4 class=41 field="SubType" optionlist=1 any=1>`

Taxonomy	Lists the IDs of taxonomy words. Example: `<pbs:!ccefields module=4 field="Taxonomy" optionlist=1 any=1>`
Type (category)	Lists the categories defined for the class. Example: `<pbs:!ccefields module=4 class=41 field="Type" optionlist=1 any=1>`
Vertical	Adding the option vertical to the tag makes the tag list the options vertically instead of horizontally. The option can only be used with the **Columns** option. The option is used like this: `Vertical=1`
Week	Displays the dates for the days in the last week.

CCE_ModuleShortNameFieldsFieldName.pbo

Example file name: CCE_WeatherFieldsType.pbo

<%Items%>	Outputs the items.Example: `<table border=1 cellpadding=1 cellspacing=1><%Items%></table>`

CCE_ModuleShortNameFieldsFieldNameItem.pbo

This file is output per item.
Example file name: cce_weatherFieldsTypeItem.pbo

<%Id%>	The ID for the item.

<%Name%>	The name for the item. The fields can be used together like this:
	
	`<option value="<%Id%>"><%Name%></option>`
	
	Or like this:
	
	`<tr><td><%Id%></td><td><%Name%></td></tr>`

<%Value%>	Returns the data type defined for a custom field (see above).

CCE_ModuleShortNameFieldsFieldNameRow.pbo

When you use the **Columns** option, you must use the cce_ModuleShortNameFieldNameRow.pbo file.

Example file name: cce_weatherFieldsTypeRow.pbo

<%Col1%>	Outputs the first row.
<%Col2....9%>	You will have a col field for each row.

pbs LastUpdate for CCE

<pbs:!#LastUpdate#>

Outputs the update time for the category being viewed.

pbs usersubmitform

> Use this link. To see information on how you can submit Stories and Galleries.
>
> https://docs.newscyclesolutions.com/display/Onl/pbs+usersubmitform+for+UGC
>
> The below documentation might be lacking information

<pbs:usersubmitform>

The tag is used to submit data to the system through the front-end. This section will focus on submitting CCE objects.

Every CCE object specific form tag has to specify a specific Module, Class, and Type that the CCE object will belong to.

ObjectClass	Specifies what object file design class to use.
CCESection	Specifies Module, Class, and Type that the CCE object will belong to on the form "<ModuleID><*ClassID*><TypeID>". Matching settings in the publicus.ini file are required for the user to be able to do a succesful submit (see https://docs.newscyclesolutions.com/display/Onl/UserSubmitCCE+Section). The <pbs:UserSubmitForm> tag will appear in the source of the page unless the correct section exists in the publicus.ini file.
IniSectionPostfix	This will allow the system to use a different UserSubmit ini section for the posting rules. This means that it is possible to have anonymous post in some forms on the site, while not allowing it in others. Be sure to protect the ini sections that allow anonymous post with categories or ForceStatus. If an ini setting is not found in UserSubmit_Postfix, it will default back to the UserSubmit section.

SubmitType	Specifies what type of data is to be submitted. For CCE objects this will be 'cce'. Default is 'ar' for article.

Example:

```
<pbs:usersubmitform ObjectClass="1" CCESection="2_12_Cars" SubmitType="CCE">
```

User Generated Content Tags

This function allows a web guest to publish a story directly into the SAXOTECH Online database. This feature can be set up to either require registration or login. The stories will appear in the editing interface unpublished on the date the form was submitted. Use the SAXOTECH Online interface to manage the stories.

- User Submitted Stories
- pbs userarticleform
  - ArticleForm.pbo
  - ArticleFormMediaItem.pbo
  - ArticleFormSectionItem.pbo
  - UserSubArticles_ArticleFormNotLoggedIn
- pbs usersubmitform for UGC
  - UserSubmitForm.pbo
  - UserSubmitFormArticle.pbo
  - UserSubmitFormArticleCategoryList.pbo
    - UserSubmitFormArticleCategoryListItem.pbo
  - UserSubmitFormArticleErrorFailed.pbo
  - UserSubmitFormArticleErrorInvalidInput.pbo
  - UserSubmitFormArticleErrorNotCreator.pbo
  - UserSubmitFormArticleErrorNotLoggedIn.pbo
  - UserSubmitFormArticleErrorNoUGCAccess.pbo
  - UserSubmitFormArticleExtraFields.pbo
    - UserSubmitFormArticleExtraFieldsItem.pbo
  - UserSubmitFormArticleMediaItem.pbo
  - UserSubmitFormArticleSectionItem.pbo
  - UserSubmitFormArticleTaxonomyList.pbo
    - UserSubmitFormArticleTaxonomyListItem.pbo
  - UserSubmitFormCategoryList.pbo
    - UserSubmitFormCategoryListItem.pbo
  - UserSubmitFormCCE.pbo
  - UserSubmitFormCCECategoryList.pbo
    - UserSubmitFormCCECategoryListItem.pbo
  - UserSubmitFormCCEErrorFailed.pbo
  - UserSubmitFormCCEErrorInvalidInput.pbo
  - UserSubmitFormCCEErrorNotCreator.pbo
  - UserSubmitFormCCEErrorNotLoggedIn.pbo
  - UserSubmitFormCCEErrorNoUGCAccess.pbo
  - UserSubmitFormCCEFieldList.pbo
    - UserSubmitFormCCEFieldListItem.pbo
  - UserSubmitFormCCEModule.pbo
  - UserSubmitFormCCERecurrencePattern.pbo
  - UserSubmitFormCCETaxonomyList.pbo
    - UserSubmitFormCCETaxonomyListItem.pbo
  - UserSubmitFormErrorFailed.pbo
  - UserSubmitFormErrorInvalidInput.pbo
  - UserSubmitFormErrorNotCreator.pbo
  - UserSubmitFormErrorNotLoggedIn.pbo
  - UserSubmitFormErrorNoUGCAccess.pbo
  - UserSubmitFormExtraFields.pbo
    - UserSubmitFormExtraFieldsItem.pbo
  - UserSubmitFormPhotoGallery.pbo
  - UserSubmitFormPhotoGalleryCategoryList.pbo
    - UserSubmitFormPhotoGalleryCategoryListItem.pbo
  - UserSubmitFormPhotoGalleryErrorFailed.pbo
  - UserSubmitFormPhotoGalleryErrorInvalidInput.pbo
  - UserSubmitFormPhotoGalleryErrorNotCreator.pbo
  - UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo
  - UserSubmitFormPhotoGalleryErrorNoUGCAccess.pbo
  - UserSubmitFormPhotoGalleryExtraFields.pbo
    - UserSubmitFormPhotoGalleryExtraFieldsItem.pbo
  - UserSubmitFormPhotoGalleryPhotoItem.pbo
  - UserSubmitFormPhotoGalleryTaxonomyList.pbo
    - UserSubmitFormPhotoGalleryTaxonomyListItem.pbo
  - UserSubmitFormTaxonomyList.pbo
    - UserSubmitFormTaxonomyListItem.pbo

From version 6.2.0 the feature has been completely reworked. The existing object files have been renamed, giving them a more general prefix because it is now possible to submit items other than stories. The system will fallback to the pre-6.2.0 object filenames for backward compatibility. However, they should be considered deprecated.

\

- ArticleForm.pbo Renamed to UserSubmitFormArticle.pbo
- ArticleFormMediaItem.pbo Renamed to UserSubmitFormArticleMediaItem.pbo
- ArticleFormNotLoggedIn.pbo Renamed to UserSubmitFormArticleErrorNotLoggedIn.pbo
- ArticleFormSectionItem.pbo Renamed to UserSubmitFormArticleSectionItem.pbo

- <pbs:userarticleform> Renamed to <pbs:UserSubmitForm>

The new section

The tag <Pbs:UserSubmitForm> generates a submit form formatted through a number of object files and can load in an existing item for updating purposes.

The object files are listed below and they allow specialization through the filename. For example, generating a story submit form will first try to retrieve the file prefix *UserSubmitFormArticle*. If it is not available, it will try the file prefix *UserSubmitForm*. Finally, if applicable it will try the depricated file above.

The following object files can display stories, photo galleries, and CCE objects:

UserSubmitForm.pbo
UserSubmitFormTaxonomyList.pbo
UserSubmitFormTaxonomyListItem.pbo

UserSubmitFormErrorNotLoggedIn.pbo
UserSubmitFormErrorNotCreator.pbo
UserSubmitFormErrorInvalidInput.pbo
UserSubmitFormErrorFailed.pbo

The following general object files are only applicable for displaying stories and photo galleries:

UserSubmitFormCategoryList.pbo
UserSubmitFormCategoryListItem.pbo
UserSubmitFormExtraFields.pbo
UserSubmitFormExtraFieldsItem.pbo
Live-Extrafields\<FieldName>.pbo

The following object files are available for stories:

UserSubmitFormArticle.pbo
UserSubmitFormArticleCategoryList.pbo
UserSubmitFormArticleCategoryListItem.pbo
UserSubmitFormArticleTaxonomyList.pbo
UserSubmitFormArticleTaxonomyListItem.pbo
UserSubmitFormArticleExtraFields.pbo
UserSubmitFormArticleExtraFieldsItem.pbo
UserSubmitFormArticleMediaItem.pbo
UserSubmitFormArticleSectionItem.pbo

UserSubmitFormArticleErrorNotLoggedIn.pbo
UserSubmitFormArticleErrorNotCreator.pbo
UserSubmitFormArticleErrorInvalidInput.pbo
UserSubmitFormArticleErrorFailed.pbo

The following object files are available for photo galleries:

UserSubmitFormPhotoGallery.pbo
UserSubmitFormPhotoGalleryCategoryList.pbo
UserSubmitFormPhotoGalleryCategoryListItem.pbo
UserSubmitFormPhotoGalleryTaxonomyList.pbo
UserSubmitFormPhotoGalleryTaxonomyListItem.pbo
UserSubmitFormPhotoGalleryExtraFields.pbo
UserSubmitFormPhotoGalleryExtraFieldsItem.pbo
UserSubmitFormPhotoGalleryPhotoItem.pbo

UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo
UserSubmitFormPhotoGalleryErrorNotCreator.pbo
UserSubmitFormPhotoGalleryErrorInvalidInput.pbo
UserSubmitFormPhotoGalleryErrorFailed.pbo

The following object files are available for CCE objects:

UserSubmitFormCCE.pbo
UserSubmitFormCCECategoryList.pbo
UserSubmitFormCCECategoryListItem.pbo

UserSubmitFormCCETaxonomyList.pbo
UserSubmitFormCCETaxonomyListItem.pbo
UserSubmitFormCCEModule.pbo
UserSubmitFormCCEFieldList.pbo
UserSubmitFormCCEFieldListItem.pbo
UserSubmitFormCCERecurrencePattern.pbo

UserSubmitFormCCEErrorNotLoggedIn.pbo
UserSubmitFormCCEErrorNotCreator.pbo
UserSubmitFormCCEErrorInvalidInput.pbo
UserSubmitFormCCEErrorFailed.pbo

User Submitted Stories

This function allows a web guest to publish a story directly into the SAXOTECH Online database. This feature can be set up to either require registration or login. The stories will appear in the editing interface unpublished on the date the form was submitted. Use the SAXOTECH Online interface to manage the stories.

The following object files can display stories:

UserSubmitForm.pbo
UserSubmitFormTaxonomyList.pbo
UserSubmitFormTaxonomyListItem.pbo
UserSubmitFormErrorNotLoggedIn.pbo
UserSubmitFormErrorNotCreator.pbo
UserSubmitFormErrorInvalidInput.pbo
UserSubmitFormErrorFailed.pbo

The following general object files are also available for displaying stories:

UserSubmitFormCategoryList.pbo
UserSubmitFormCategoryListItem.pbo
UserSubmitFormExtraFields.pbo
UserSubmitFormExtraFieldsItem.pbo
Live-Extrafields\<FieldName>.pbo

The following object files are available for stories:

UserSubmitFormArticle.pbo
UserSubmitFormArticleCategoryList.pbo
UserSubmitFormArticleCategoryListItem.pbo
UserSubmitFormArticleTaxonomyList.pbo
UserSubmitFormArticleTaxonomyListItem.pbo
UserSubmitFormArticleExtraFields.pbo
UserSubmitFormArticleExtraFieldsItem.pbo
UserSubmitFormArticleMediaItem.pbo
UserSubmitFormArticleSectionItem.pbo

UserSubmitFormArticleErrorNotLoggedIn.pbo
UserSubmitFormArticleErrorNotCreator.pbo
UserSubmitFormArticleErrorInvalidInput.pbo
UserSubmitFormArticleErrorFailed.pbo

pbs userarticleform

<pbs:userarticleform>

This tag is used to display the form.

A form is required for users to submit stories. The form requires that the user is logged in. You can either set the security option on the category by setting the subscription type on the category, or you can set the category to require registration by adding `&Login=1` to the link.

Template

This function should be used on a section template. The example is placed on Usersubmit_Ovr.pbs.

Options

Mediafiles	Allows multiple asset files to be submitted with a story. Enter `mediafiles=x` where x indicates the number of possible assets that can be uploaded.
Objectclass	Sets the object class used.

Sections	Controls how many paragraphs the page will contain. By default, 7 paragraph boxes are displayed. Enter
	`sections=x`
	to have
	`x`
	number of paragraph boxes on the page.

Useobjects	Enables the use of object files.

ArticleForm.pbo

Example code:

```
<form action="<%PostAction%>" method="POST" enctype="multipart/form-data">
<input type="hidden" name="status" value="1">

*** Input Fields ***

<input type="submit" value="submit">
</form>
```

The object file contains the following fields:

Byline	Writes the byline to the database. It is used like this: `<input type="text" size="50" name="byline">`
UTF-8	Translate text to UFT-8 `<input name="UTF-8" value="1" type="hidden">`

Category	Determines the category in which the story will be saved. It is used like this:
	
	`<select name="category">`
	`<option value="LETTERS">Letters`
	`<option value="OPINION">Opinion`
	`</select>`

<%Counter%>	The counter for the paragraphs.
<%Date%>	The current date. It can be used in conjunction with the date input field like this: `<b>Date:</b> <input type="text" size="10" name="date" value="<%Date%>">`
Forum	Set forum or usercomment informations for the story. Default is 0: `<input type="hidden" name="forum" value="1">` 0 = Off 1 = On 2 = Locked
Location	Writes the location to the database. It is used like this: `<input type="text" size="30" name="location">`
<%mediafiles%>	Outputs the media files, such as images.
Picture	The local path to a picture that will be uploaded. It is used like this: `<input type="file" size="30" name="picture" value=" Picture... ">`
Picturebyline	The picture byline. `<input type="text" size="50" name="picturebyline">`

Picturecaption	The picture caption. `<textarea cols="50" rows="3" name="picturecaption"></textarea> `
Picturecaptionheading	The picture caption heading. `<input type="text" size="50" name="picturecaptionheading">`
<%PostAction%>	Contains the URL used by the submission form. It is used like this: `<form action="<%PostAction%>" method="POST" enctype="multipart/form-data"> <input type="hidden" name="status" value="1">`
RedirectURL	Sets the redirection page to which the web guest is sent after submitting the form. The field is used like this: `<input type="hidden" name="RedirectURL" value="/apps/pbcs.dll/section?category=SUB">`
<%Sections%>	Outputs the paragraphs.
Status	This field can be used to override the story status. By default, stories come in as unpublished. If you would like to publish automatically upon submission, enter the following line in the form: `<input type="hidden" name="status" value="9">`
Subtitle	Writes the subtitle to the database. It is used like this: `<input type="text" size="66" name="title2">`
Summary	Writes the summary to the database. It is used like this: `<textarea cols="50" rows="7" name="summary"></textarea>`

Superheading	Writes the Superheading to the database. It is used like this: `<input type="text" size="66" name="superheading">`
Nettitle	Writes the net title to the database. It is used like this: `<input type="text" size="66" name="webheading">`
Title	Writes the title to the database. It is used like this: `<input type="text" size="66" name="title">`
<%UserEmail%>	The logged in user's email address.
<%UserName%>	The logged in user's first and last name.

ArticleFormMediaItem.pbo

The ArticleFormMediaItem.pbo is used when rendering the submit form. It must contain the following:

name="media<%Counter%>"	Selects the media file.
name="media<%Counter%>description"	Used for a description of the media.

The object file has the following pbo variables:

<%counter%>	Returns the number of assets.
<%NoOfSections%>	Returns the value of `<pbs:userarticleform>` tag option "mediafiles."

ArticleFormSectionItem.pbo

The object file contains the following fields:

body<%Counter%>	The paragraph body. It is used like this: `<textarea cols="50" rows="7" name="body<%Counter%>"></textarea>`
body<%Counter%>image	This selects the image file to be used with the paragraph. It is used like this:

body<%Counter%>imagebyline	The byline to be used with the selected image in the paragraph. It is used like this:
body<%Counter%>imagecaption	The caption to be used with the selected image in the paragraph. It is used like this:
body<%Counter%>imagetext	Any text to be used with the selected image in the paragraph. It is used like this:
intermediateheading<%Counter%>	The paragraph heading. It is used like this: `<input type="text" size="66" name="intermediateheading<%Counter%>" >`

This ArticleFormSectionItem.pbo file has the following pbo variables:

<%counter%>	Returns the paragraph number
<%NoOfSections%>	This variable returns the value of the `<pbs:userarticleform>` tag option "sections."

UserSubArticles_ArticleFormNotLoggedIn

ArticleFormNotLoggedIn.pbo

This object file contains no fields.

pbs usersubmitform for UGC

<pbs:usersubmitform>

The tag is used to submit data to the system through the front-end. The data can be a story, a photo gallery or a CCE object, and the data type is determined through the parameter of **SubmitType**.

Parameters available

ObjectClass	Specifies what object file design class to use.
SubmitType	Specifies what type of data is to be submitted. Use 'ph' for photo gallery, 'cce' for CCE object, and 'ar' for story. Default is 'ar'.

Stories

Every story specific form tag supports the following parameters

Sections	Defines the maximum number of paragraphs to be available to the user.
MediaFiles	Defines the maximum number of media files that can be uploaded by the user.

Photo galleries

Every gallery specific form tag supports the following parameters

Sections	Same as **MaxAllowedPhotos**.

MaxAllowedPhotos	Defines the maximum number of images to be available to the user.

CCE objects

Every CCE object specific form tag has to specify a specific Module, Class, and Type that the CCE object will belong to.

CCESection	Specifies Module, Class, and Type that the CCE object will belong to on the form "<ModuleID><*ClassID*><TypeID>". Matching settings in the publicus.ini file are required for the user to be able to do a succesful submit. See here: https://docs.newscyclesolutions.com/display/Onl/UserSubmitCCE+Section

Matching settings in the publicus.ini file are required for the user to be able to do a succesful submit. **- This must be in place for the pbs tag to actually pull in the object file.**

```
[UserSubmitCCE_3_303_all]
CCEModule=3
CCEClass=303
CCEType=all
```

```
<pbs:usersubmitform ObjectClass="1" CCESection="2_12_Cars" SubmitType="CCE">
```

Updating an existing item

If an existing item is to be loaded into the form so that the item is updated on save, the URL parameters to the call that fetches the `<pbs:UserSubmitForm>` tag contents must contain **load=1** and the **ID of the item to be loaded**.

For stories the following ID parameters are required:

```
/apps/pbcs.dll/section?category=storysubmit&OrgDate=X&OrgCategory=Y&OrgStoryID=Z&load=1
```

Where X, Y, and Z values match a story within the system.

For photo galleries the following ID parameters are required:

```
/apps/pbcs.dll/section?category=gallerysubmit&OrgDate=X&OrgCategory=Y&OrgStoryID=Z&load=1
```

Where X, Y, and Z values match a photo gallery within the system.

Update CCE item

For CCE objects the following ID parameter is required:

```
/apps/pbcs.dll/section?category=ccesubmit&OrgCCEID=x&load=1
```

Where the value of X matches a CCE object by GUID within the system, like:

```
/apps/pbcs.dll/section?category=ccesubmit&OrgCCEID=E31B293F-0A49-45A5-ADDC-5E6E6A606679&load=1
```

On the CCE Item object file:

```
/apps/pbcs.dll/section?category=ccesubmit&OrgCCEID=<%OrgCCEID%>&load=1
```

UserSubmitForm.pbo

Object file for formatting the submit form for the user generated by <pbs:usersubmitform>.The object files of UserSubmitFormArticle.pbo, UserSubmitFormPhotoGallery.pbo, and UserSubmitFormCCE.pbo are also available for the specific **SubmitType**.

Form	<form name="StorySubmit" method="POST" enctype="multipart/form-data" action="<%PostAction%>">
RedirectURL	<input type="hidden" name="RedirectURL" value="/section/success">
UTF-8	<input name="UTF-8" value="1" type="hidden">

The form's action url contains an URL parameter, **SubmitMode**, which is either 'save', 'delete', 'update' or 'append'. Save is for creating a new item, Update and Append are for modifying an exisiting, and Delete is for deleting an item. The difference between Update and Append is that Update signifies that the whole item is to be updated, whereas Append means that only the sections, i.e., paragraphs and images, are to be updated. The SubmitMode is added by the system in the various specific links output in object files. In addition, the **AppendSections** field described later allows for truly appending new paragraphs and images without overwriting the existing ones. Examples of a form tag:

```
<form action="<%PostAction%>" method="POST" enctype="multipart/form-data">
<form action="<%PostAction$r("update=append")%>" method="POST"
enctype="multipart/form-data">
```

Where the first form header indicate the original PostAction it will when editing an existing item do a full overwrite as the system adds the URL parameter "SubmitMode=update" automatically. The second example actively replaces that parameter value to be "append".

The following general variables are available for all submit types:

<%Date%>	Contains the current date on the form YYYYMMDD.
<%PostAction%>	Contains the URL the form is to be submitted to.
<%SubmitType%>	Contains what kind of content item the form is to submit.
<%TaxonomyWords%>	Contains the taxonomy words available to the user formatting using the object file of UserSubmitFormTaxonomyList.pbo
<%UserID%>	The user's ID. If the user is anonymous the value matches the AnonymousUserId setting.
<%UserName%>	The user's full name, e.g., John Doe. If the user is anonymous the variable is empty.
<%UserEmail%>	The user's email address. If the user is anonymous the variable is empty.

Also, there are Story specific, Photo gallery specific, and CCE specific variables.

Story submit form

AppendSections **Version 7.2.0sr17**	Specifies if all sections should be appended. Default is 0. If this is 0 all sections will be saved at their current location. <input type="hidden" name="AppendSections" value="1">

Story specific variables

<%Categories%>	Contains the available categories formatted using the object file of UserSubmitFormCategoryList.pbo. This is controlled by an publicus.ini setting: Categories on user generated content
<%ExtraFields%>	Contains the extra fields formatted using the object file of UserSubmitFormExtraFields.pbo
<%ExtraFieldType%>	Contains the name of the extra fields' group name.
<%MediaFiles%>	Contains the media files input fields formatted using the object file of UserSubmitFormArticleMediaItem.pbo
<%NoOfMediaFiles%>	Contains the number of allowed media files in the form based on the tag parameter of **MediaFiles**.
<%Sections%>	Contains the story paragraphs formatted using UserSubmitFormArticleSectionItem.pbo
<%NoOfSections%>	Contains the number of paragraphs.
<%Forum%>	Contains the forum value of the story. 0 is off, 1 is on, 3 is locked.
<%OutputEmptySections%> **Version 7.2.0sr17**	This variable is 1 if empty sections are generated in the sections variable. See publicus.ini section UserSubmit to change this.

When updating an existing story, the following variables are available as well:

<%Status%>	Contains the published status of the story. 1 is unpublished, 9 is published.
<%ArticleStatus%>	Same as **<%Status%>**
<%Summary%>	Contains the summary of the story.
<%Title%>	Contains the title of the story.
<%Location%>	Contains the location of the story.

<%PictureCaption%>	Contains the caption of the main image.
<%PictureSummary%>	Contains the image summary of the main image.
<%PictureByline%>	Contains the image byline of the main image.
<%PictureURL%>	Contains a link to the main image.

<%OrgSections%>	Contains the number of paragraphs in the original.
<%OrgDate%>	Contains the original date.
<%OrgStoryID%>	Contains the original ID.
<%OrgCategory%>	Contains the original category.

Photo gallery submit form

AppendSections **Version 7.2.0sr17**	Specifies if all sections should be appended. Default is 0. If this is 0 all sections will be saved at their current location. <input type="hidden" name="AppendSections" value="1">

Photo gallery specific variables

<%Categories%>	Contains the available categories formatted using the object file of UserSubmitFormCategoryList.pbo
<%ExtraFields%>	Contains the extra fields formatted using the object file of UserSubmitFormExtraFields.pbo
<%ExtraFieldType%>	Contains the name of the extra fields' group name.
<%Sections%>	Contains the gallery images formatted using UserSubmitFormPhotoGalleryPhotoItem.pbo
<%NoOfSections%>	Contains the number of images.
<%OutputEmptySections%> **Version 7.2.0sr17**	This variable is 1 if empty sections are generated in the sections variable. See publicus.ini section UserSubmit to change this.

When updating an existing gallery, the following variables are available as well:

<%Status%>	Contains the published status of the gallery. 0 is unpublished, 1 is published.
<%PhotoGalleryStatus%>	Same as **<%Status%>**
<%Summary%>	Contains the summary of the gallery.
<%Title%>	Contains the title of the gallery.
<%Location%>	Contains the location of the gallery.

<%OrgSections%>	Contains the number of images in the original.
<%OrgDate%>	Contains the original date.
<%OrgStoryID%>	Contains the original ID.
<%OrgCategory%>	Contains the original category.

CCE Submit form

Module	<input type="hidden" name="Module" value="<%Module%>">
Class	<input type="hidden" name="Class" value="<%Class%>">
Type	<input type="hidden" name="CCEType" value="<%CCEType%>">
Title	<input type="text" name="Title" value="<%Title%>">
Text	<input type="text" name="Text" value="<%Text%>">
TypesList	<input type="hidden" name="TypesList" value="TypeID1,Typeid2">

CCE object specific variables

<%Module%>	Contains the module and type information formatted using the object file of UserSubmitFormCCEModule.pbo
<%ShowRecurrencePattern%>	Contains whether the object to be submitted belongs to an Event module (=1) or not (=0)

<%Fields%>	Contains the user defined fields formatted using the object file of **UserSubmitFormCCEFieldList.pbo**
<%Field_[FieldName]%>	Contains the value of the user defined field. The value of all user defined fields are available in the main object file UserSubmitForm.pbo e.g. <%field_Price%> will contain the value of user defined field "price"

If **<%ShowRecurrencePattern%>** is 1 then the following variables are available as well:

<%CCERecurrencePattern%>	Contains the recurrence pattern formatted using the object file of UserSubmitFormCCERecurrencePattern.pbo
<%Pattern%>	Contains the overall recurrence pattern used for calculating when the object event occurs.
<%StopMethod%>	Contains the method used for determining when the event series stops. 0 means a set date, 1 means after a certain amount of occurrences.

If editing an existing object, the following variables are available:

<%Expires%>	Contains the expire date of the object.
<%Latitude%>	Contains the latitude of the object.
<%Longitude%>	Contains the longitude of the object.
<%PictureURL%>	Contains the image link to the object's attached image if there is one.
<%LogoUrl%> Version 7.0.0sr11	Contains the logo link to the object's attached logo image if there is one.
<%RowGUID%>	The ID of the object.
<%Summary%>	Contains the summary of the object.
<%Title%>	Contains the title of the object.

Story Submit Form Example:

```
<form name="StorySubmit" method="POST" enctype="multipart/form-data"
action="<%PostAction%>">

<input type="hidden" name="status" value="1"/>
<input type="hidden" name="html" value="1"/>
<input type="hidden" name="ExtraFieldType" value="<%ExtraFieldType%>"/>
<input type="hidden" id="Taxonomy" name="Taxonomy" value=""/>
<input type="hidden" id="DeleteSection" name="DeleteSection" value=""/>
<input type="hidden" id="DeletedImages" name="DeletedImages" value=""/>


<input type="hidden" name="OrgStoryID" value="<%OrgStoryID%>"/>
<input type="hidden" name="OrgDate" value="<%OrgDate%>"/>
<input type="hidden" name="OrgCategory" value="<%OrgCategory%>"/>
<input type="hidden" name="OrgSections" value="<%OrgSections%>"/>
<input type="hidden" name="SectionCount" value="<%NoOfSections%>"/>
<input type="hidden" name="Sections" value="<%NoOfSections%>"/>
<input type="hidden" class="arttext" name="date" value="<%Date%>" size=8/>
<input type="hidden" name="RedirectURL" value="/section/success"/>

<div>Forum</div>
<div>
On <input type="radio" name="forum" value="1" [%{Forum=1}checked%]/>
- Off <input type="radio" name="forum" value="0" [%{Forum=0}checked%]/>
- Locked <input type="radio" name="forum" value="3" [%{Forum=3}checked%]/>
</div>

<input type="text" name="title" value="<%Title%>" class="StoryTitle"/>
<textarea name="summary" class="StorySummary"><%Summary%></textarea>
<input type="text" name="byline" value="<%UserName%>" class="StoryByline"/>
<input type="text" name="location" value="<%location%>" class="StoryLocation"/>
<input type="file" name="picture" value="Picture..." class="MainPictureFile"/>

<input type="submit" value="Create"/>

</form>
```

Photo gallery Submit Form Example:

```
<form method="POST" enctype="multipart/form-data" action="<%PostAction%>">
<input name="UTF-8" value="1" type="hidden">
<input type="hidden" name="RedirectURL" value="/section/success"/>

<b>Photoalbum</b><br />
Title: <input type="text" name="title"><br />
Summary: <textarea name="summary" class="Summary"></textarea><br /><br />

<b>Picture 1</b><br />
Title: <input type="text" name="body1imagecaption" value="body1imagecaption"><br />
Text: <input type="text" name="body1imagetext" value="body1imagetext"><br />
Byline: <input type="text" name="body1imagebyline" value="body1imagebyline"><br />
Pictures 1: <input type="file" name="picture" value="Picture..." /><br /><br>

<b>Picture 2</b><br />
Title: <input type="text" name="body2imagecaption" value="body2imagecaption"><br />
Text: <input type="text" name="body2imagetext" value="body2imagetext"><br />
Byline: <input type="text" name="body2imagebyline" value="body2imagebyline"><br />
Pictures 2: <input type="file" name="picture2" value="Picture..." /><br /><br />

<b>Picture 3</b><br />
Title: <input type="text" name="body3imagecaption" value="body3imagecaption"><br />
Text: <input type="text" name="body3imagetext" value="body3imagetext"><br />
Byline: <input type="text" name="body3imagebyline" value="body3imagebyline"><br />
Pictures 3: <input type="file" name="picture3" value="Picture..." /><br />

<input type="submit" value="Create"/>
</form>
```

CCE Submit Form Example:

```
<form method="POST" enctype="multipart/form-data" action="<%PostAction%>">
<input name="UTF-8" value="1" type="hidden">
<input type="hidden" name="RedirectURL" value="/section/success"/>
<input type="hidden" name="Module" value="2345"/>
<input type="hidden" name="Class" value="54363"/>
<input type="hidden" name="CCEType" value="DavLevel0"/>
Title: <input type="text" name="Title" value="<%Title%>"/><br />
Summary: <input type="text" name="Summary" value="<%Summary%>"/>
<input type="submit" value="Create"/>
</form>
```

Update existing photo gallery Example:

Request: /section/mojogallery&OrgDate=20120302&OrgCategory=Gallery&OrgStoryID=120309963&load=1

Before you can update and existing photo gallery you need to have this ini setting actived for the Edit site: ShowPhotoAlbumSharedStatus=1 Default is off. GENERAL section

```
<form method="POST" enctype="multipart/form-data" action="<%PostAction%>">
<input name="UTF-8" value="1" type="hidden">
<input type="hidden" name="RedirectURL" value="/section/success"/>
<input type="hidden" name="OrgDate" value="<%OrgDate%>"/>
<input type="hidden" name="OrgCategory" value="<%Orgcategory%>"/>
<input type="hidden" name="OrgStoryID" value="<%OrgStoryID%>"/>

<b>Photoalbum</b><br />
Title: <input type="text" name="title" value="<%Title%>"><br />
Summary: <textarea name="summary" class="Summary"><%Summary%></textarea><br /><br />
<%Sections%>
<input type="submit" value="Update"/>
</form>
```

Update existing CCE element Example:

Request: /section/usersubmitcce&OrgCCEID=E5DBCECA-FFFD-4F9D-B209-60B721725772&load=1

```
<form method="POST" enctype="multipart/form-data" action="<%PostAction%>">
<input name="UTF-8" value="1" type="hidden">
<input type="hidden" name="RedirectURL" value="/section/success"/>
<input type="hidden" name="Module" value="99"/>
<input type="hidden" name="Class" value="111"/>
<input type="hidden" name="Type" value="101010"/>
<input type="hidden" name="OrgCCEID" value="<%RowGUID%>"/>
Title: <input type="text" name="Title" value="<%Title%>"/><br />
Summary: <input type="text" name="Summary" value="<%Summary%>"/><br />
FromDate: <input type="text" name="FromDate" value="20120329"/><br />
ToDate: <input type="text" name="ToDate" value="20120329" /><br />

<b>Pictures</b><br />
Image01: <input type="file" name="Image01"><br />
Image02: <input type="file" name="Image02"><br />
Image03: <input type="file" name="Image03"><br />
Logo: <input type="file" name="Logo"><br /><br />


<!-- set the following hidden fields to 1 for each image if the form is supposed to
replace/delete the image in question -->
<input type="hidden" name="Img_del01" value="1" />
<input type="hidden" name="Img_del02" value="1" />
<input type="hidden" name="Img_del03" value="1" />
<input type="hidden" name="Logo_del" value="1" />
<input type="submit" value="Update"/>
</form>
```

UserSubmitFormArticle.pbo

Works the same way as UserSubmitForm.pbo.

UserSubmitFormArticleCategoryList.pbo

Works the same way as UserSubmitFormCategoryList.pbo.

UserSubmitFormArticleCategoryListItem.pbo

Works the same way as UserSubmitFormCategoryListItem.pbo.

UserSubmitFormArticleErrorFailed.pbo

Works the same way as UserSubmitFormErrorFailed.pbo.

UserSubmitFormArticleErrorInvalidInput.pbo

Works the same way as UserSubmitFormErrorInvalidInput.pbo.

UserSubmitFormArticleErrorNotCreator.pbo

Works the same way as UserSubmitFormErrorNotCreator.pbo.

UserSubmitFormArticleErrorNotLoggedIn.pbo

Works the same way as UserSubmitFormErrorNotLoggedIn.pbo.

UserSubmitFormArticleErrorNoUGCAccess.pbo

Works the same way as UserSubmitFormErrorNoUGCAccess.pbo.

UserSubmitFormArticleExtraFields.pbo

Works the same way as UserSubmitFormExtraFields.pbo.

UserSubmitFormArticleExtraFieldsItem.pbo

Works the same way as UserSubmitFormExtraFieldsItem.pbo.

UserSubmitFormArticleMediaItem.pbo

Object file for formatting the media files of the story submit form for the user generated by **<pbs:UserSubmitForm SubmitType="ar">**.

The form requires the two fields of:

name="media<%Counter%>"	File selector holding a reference to a local media file for upload.
name="media<%Counter%>description"	Used for a description of the media file.

When updating an existing story the following variables are available.

<%Counter%>	A forthrunning number.
<%NoOfMediaFiles%>	Contains the number of media files allowed by the **MediaFiles** tag parameter.

It is unfortunately not currently possible to update or delete an existing story's media files.

UserSubmitFormArticleSectionItem.pbo

Object file for formatting the story paragraphs of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="ar">**.

<%Counter%>	A forthrunning number.
<%NoOfSections%>	Contains the maximum number of paragraphs available, determined by the **Sections** tag parameter.
<%OrgSections%>	Contains the number of paragraphs on the original story when updating it - otherwise 0.
<%HasContent%>	Determines whether the paragraphs contains an image (=1) or not (=0).

In addition, if updating a story the following variables are available:

<%Body%>	Contains the paragraph text.
<%PictureCaption%>	Contains the paragraph image title.
<%PictureByline%>	Contains the paragraph image byline.

<%PictureSummary%>	Contains the paragraph image text.
<%PictureUrl%>	Contains the URL to the paragraph image.

Example:

```
Body:
<textarea cols="70" rows="9" name="body<%Counter%>"
id="body<%Counter%>"><%body%></textarea>

Image:
<input type="file" size="61" name="body<%Counter%>image" value="Picture...">

Image caption:
<input size="60 type="text" name="body<%Counter%>imagecaption"
value="<%picturecaption%>">

Image text:
<textarea rows="3" cols="45"
name="body<%Counter%>imagetext"><%picturesummary%></textarea>

Image byline:
<input size="60" type="text" name="body<%Counter%>imagebyline"
value="<%picturebyline%>">
```

UserSubmitFormArticleTaxonomyList.pbo

Works the same way as UserSubmitFormTaxonomyList.pbo.

UserSubmitFormArticleTaxonomyListItem.pbo

Works the same way as UserSubmitFormTaxonomyListItem.pbo.

UserSubmitFormCategoryList.pbo

Object file for formatting the category list part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files UserSubmitFormArticleCategoryList.pbo and UserSubmitFormPhotoGalleryCategoryList.pbo are also available for the specific **SubmitType**.

<%Items%>	Contains the available categories formatted using UserSubmitFormCategoryListItem.pbo
<%Count%>	Contains the amount of categories available.

UserSubmitFormCategoryListItem.pbo

Object file for formatting the category part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files UserSubmitFormArticleCategoryListItem.pbo and UserSubmitFormPhotoGalleryCategoryListItem.pbo are also available for the specific **SubmitType**.

<%ID%>	Contains the category ID.
<%Count%>	Forthrunning number.
<%Name%>	Contains the display text.
<%Selected%>	Has the value 'selected' if the item has the category as category. Only available when updating existing data.

UserSubmitFormCCE.pbo

Works the same way as UserSubmitForm.pbo.

UserSubmitFormCCECategoryList.pbo

Works the same way as UserSubmitFormCategoryList.pbo.

UserSubmitFormCCECategoryListItem.pbo

Works the same way as UserSubmitFormCategoryListItem.pbo.

UserSubmitFormCCEErrorFailed.pbo

Works the same way as UserSubmitFormErrorFailed.pbo.

UserSubmitFormCCEErrorInvalidInput.pbo

Works the same way as UserSubmitFormErrorInvalidInput.pbo.

UserSubmitFormCCEErrorNotCreator.pbo

Works the same way as UserSubmitFormErrorNotCreator.pbo.

UserSubmitFormCCEErrorNotLoggedIn.pbo

Works the same way as UserSubmitFormErrorNotLoggedIn.pbo.

UserSubmitFormCCEErrorNoUGCAccess.pbo

Works the same way as UserSubmitFormErrorNoUGCAccess.pbo.

UserSubmitFormCCEFieldList.pbo

Object file for formatting the CCE fields part of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**.

The following variables are available:

<%Items%>	Contains the fields, formatted using the object file UserSubmitFormCCEFieldListItem.pbo
<%NoOfFields%>	Contains the number of fields in **<%Items%>**.

UserSubmitFormCCEFieldListItem.pbo

Object file for formatting the CCE fields of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**.

The following variables are available:

<%Counter%>	Contains the the field number, starting from 1.
<%DataType%>	Contains the data type of the field.
<%DisplayName%>	Contains the display name.
<%Name%>	Contains the internal name. When using this for building the input field for a custom CCE field, the name must be preceded by 'field_'. `<input type="text" name="field_<%Name%>" value="<%Value%>">`
<%Value%>	Contains the value of the field when updating an existing object.

UserSubmitFormCCEModule.pbo

Object file for formatting the module and type information part of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**.

The following variables are available:

<%ModuleID%>	The module ID.
<%ModuleName%>	The module name.

<%ModuleShortName%>	The module short name.

If the CCE object belongs to a class, the following variables are available:

<%ClassID%>	The class ID.
<%ClassName%>	The class name.

If the CCE object belongs to a type, the following variables are available:

<%TypeID%>	The type ID.
<%TypeName%>	The type name.

UserSubmitFormCCERecurrencePattern.pbo

Object file for formatting the recurrence pattern part of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="cce">**. It is only used for objects belonging to Event modules.

The following variables are available when updating a CCE event with a calculated recurrence pattern:

<%FromDate%>	Contains from what date the system is to calculate occurrences (yyyymmdd format)
<%ToDate%>	Contains to what date the system is to calculate occurrences (yyyymmdd format) - it is used when **StopMethod** is 0.
<%NumberOfOcc%>	Contains how many occurrences to calculate from the **FromDate** - it is used when **StopMethod** is 1.
<%StopMethod%>	Contains how the system is to stop calculating occurrences. 0 is on a set date, 1 is after a certain number of occurrences.
<%FromTime%>	Contains from what time of day the object occurs.
<%ToTime%>	Contains to what time of day the object occurs.
<%Pattern%>	Contains the overall method used to calculate occurrences.
<%Subpattern%>	Contains the specific method used to calculate occurrences.
<%Jump1%>	Contains the first jump value in the pattern. E.g. 1 in "The first Monday in June".
<%Jump2%>	Contains the second jump value in the pattern. E.g., 6 in "The first Monday in June".
<%Day1%>	If Sunday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day2%>	If Monday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day3%>	If Tuesday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day4%>	If Wednesday´ is selected in the pattern then it has value 1, otherwise it has no value.
<%Day5%>	If Thurday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day6%>	If Friday is selected in the pattern then it has value 1, otherwise it has no value.
<%Day7%>	If Saturday is selected in the pattern then it has value 1, otherwise it has no value.

The following variables are available when updating a CCE object with a manual recurrence pattern (from version 6.9.0sr1):

<%Pattern%>	Contains the overall method used to calculate occurrences.

<%CustomDates%>	A comma-list of the occurrence dates.
<%CustomFromTimes%>	A comma-list of the occurrence from times.
<%CustomToTimes%>	A comma-list of the occurrence to times.

Recurrence pattern description

The pattern used to calculate object occurrences is made up by a main pattern and then a subpattern, for our purposes lets denote the full pattern by the **<%Pattern%>**,**<%SubPattern%>** values. Then the following patterns are available:

0 - Manual pattern

0 = The occurrences are described in the three comma lists of <CustomDates>, <CustomFromTimes>, <CustomToTimes>.

1 - Daily pattern

1,1 = For every <Jump1> days
1,2 = Every weekday (from Monday to Friday)

2 - Weekly pattern

2,1 = Repeat for every <Jump1> weeks on <DayX,Y,Z>

where <DayX,Y,Z> is actually specified by having the value set to 1 for any number of the following fields:
<Day1>,<Day2>,<Day3>,<Day4>,<Day5>,<Day6>,<Day7>

3 - Monthly pattern

3,1 = On day <Jump1> every <Jump2> months
3,2 = The First..Fourth,Last(=5) (<Jump1>) <DayN> every <Jump2> months

where <DayN> is actually specified by having the value set to 1 for one of the following fields:
<Day1>,<Day2>,<Day3>,<Day4>,<Day5>,<Day6>,<Day7>

If more than one is selected, its the first selected day in the week that is the actual one being selected.

4 - Yearly pattern

4,1 = On day <Jump1> in Month <Jump2>
4,2 = The First..Fourth,Last(=5) (<Jump1>) <DayN> in Month <Jump2>

where <DayN> is actually specified by having the value set to 1 for one of the following fields:
<Day1>,<Day2>,<Day3>,<Day4>,<Day5>,<Day6>,<Day7>

If more than one is selected, its the first selected day in the week that is the actual one being selected.

Examples:

FromDate may not be the first day in the pattern, e.g., when selecting only Mondays and today is a Tuesday, the first event date is actually 6 days in the future. The following examples show how patterns selects occurrence days

1 - Daily

1,1 = For every 2 days - If **FromDate** is a Monday, then it selects Monday, Wednesday, Friday etc.

1,2 = Selection is defined as weekdays - if **FromDate** is a Thursday, it selects Thursday, Friday, Monday, Tuesday, Wednesday, Thursday etc.

2 - Weekly

2,1 = Repeat for every 2 weeks on Mondays and Tuesdays - if **FromDate** is a Tuesday, that day is selected. The next weeks' Monday and Tuesday is not selected. In the week after that week the Monday and Tuesday is selected. Then two weeks later Monday and Tuesday is selected again.

2,1 = Repeat for every 2 weeks on Mondays and Tuesdays - if **FromDate** is a Wednesday, no selections are made for the week. The next weeks' Monday and Tuesday is not selected. In the week after that week the Monday and Tuesday is selected. Then two weeks later Monday and Tuesday is selected again.

3 - Monthly

3,1 = On day 14 every 2 months - if **FromDate** is February 13th, it selects February 14th, April 14th, June 14th etc.

3,1 = On day 14 every 2 months - if **FromDate** is February 17th, it selects April 14th, June 14th etc.

3,2 = The second Tuesday every 1 month. If **FromDate** is February 17th, 2009, the second Tuesday is the 10th. Therefore, the first occurrence is the second Tuesday of March, second occurrence is the second Tuesday in April etc. Had the second Tuesday in the current month been in the future, then that Tuesday would have been the first event date.

4 - Yearly

4,1 = On day 14 in month 2 - if **FromDate** is February 17th, 2009 the first occurrence is February 14th, 2010. Had the **FromDate** been a week before, the first occurrence had been on February 14th, 2009.

4,2 = The last Sunday in month 2 - if **FromDate** is before the given date, the first occurrence is in the same year, otherwise they start being calculated from the next.

Form examples

Submit a simple, non-recurring event

Here we want a simple input form, where the event enterer only needs to supply date and time of the event.

Because we don't want any recurrence to happen, we preselect the recurrence pattern to stop after one occurrence. And the pattern we choose is the daily pattern (1,1) with jump 1 (every day) so that we know for sure that the actual event date is going to be the same day as the start date.

```
<input type="hidden" name="stopmethod" value="1" />
  <input type="hidden" name="numberofocc" value="1" />
  <input type="hidden" name="pattern" value="1" />
  <input type="hidden" name="pattern1sub" value="1" />
  <input type="hidden" name="pattern1sub1jump1" value="1" />

  <tr>
    <td>
      <table cellpadding=1 cellspacing=0 border=0>

        <tr>
          <td>
            Date of the event:
          </td>
          <td>
            <input name="fromdate" id="fromdate" maxlength=8 value="<%FromDate%>" />
(yyyymmdd)
          </td>
        </tr>

        <tr>
          <td>
            Start time:
          </td>
          <td>
            <input name="fromtime" id="fromtime" maxlength=4 value="<%FromTime%>" />
(HHMM, 24h)
          </td>
        </tr>

        <tr>
          <td>
            End time:
          </td>
          <td>
            <input name="totime" id="totime" maxlength=4 value="<%ToTime%>" /> (HHMM,
24h)
          </td>
        </tr>

      </table>
    </td>
  </tr>
```

UserSubmitFormCCETaxonomyList.pbo

Works the same way as UserSubmitFormTaxonomyList.pbo.

UserSubmitFormCCETaxonomyListItem.pbo

Works the same way as UserSubmitFormTaxonomyListItem.pbo.

UserSubmitFormErrorFailed.pbo

Object file for formatting the response where something unexpected happened.

The object files UserSubmitFormArticleErrorFailed.pbo, UserSubmitFormPhotoGalleryErrorFailed.pbo, and UserSubmitFormCCEErrorFailed.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message

UserSubmitFormErrorInvalidInput.pbo

Object file for formatting the response for the user trying to submit a form but with invalid data.

The object files of UserSubmitFormArticleErrorInvalidInput.pbo, UserSubmitFormPhotoGalleryErrorInvalidInput.pbo, and UserSubmitFormCCEErrorInvalidInput.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message
<%ErrorField%>	Contains the name of the field containing invalid data.

UserSubmitFormErrorNotCreator.pbo

Object file for formatting the response for the user trying to submit a form holding an existing item that is being updated and where the user is is not the creator of the item.

The object files UserSubmitFormArticleErrorNotCreator.pbo, UserSubmitFormPhotoGalleryErrorNotCreator.pbo, and UserSubmitFormCCEErrorNotCreator.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message.
<%ErrorField%>	Contains the value 'User'.

UserSubmitFormErrorNotLoggedIn.pbo

Object file for formatting the response for the user trying to submit a form but where the user is not logged in.

The object files of UserSubmitFormArticleErrorNotLoggedIn.pbo, UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo, and UserSubmitFormCCEErrorNotLoggedIn.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message.
<%ErrorField%>	Contains the value 'User'.

UserSubmitFormErrorNoUGCAccess.pbo

From version 6.9.0sr5

Object file for formatting the response for the user trying to submit a form but where the user is not permitted to generate content.

The object files of UserSubmitFormArticleErrorNoUGCAccess.pbo, UserSubmitFormPhotoGalleryErrorNoUGCAccess.pbo, and UserSubmitFormCCEErrorNoUGCAccess.pbo are also available for the specific **SubmitType**.

<%ErrorMessage%>	Contains an error message.
<%ErrorField%>	Contains the value 'User'.

UserSubmitFormExtraFields.pbo

Object file for formatting the extra fields list part of the submit form for the user generated by pbs usersubmitform. The object files of UserSubmitFormArticleExtraFields.pbo and UserSubmitFormPhotoGalleryExtraFields.pbo are also available for the specific **SubmitType**.

<%Items%>	Contains the extra fields formatted using **UserSubmitFormExtraFields.pbo**.
<%Count%>	Contains the amount of extra fields.

UserSubmitFormExtraFieldsItem.pbo

Object file for formatting the extra fields part of the submit form for the user generated by <pbs:UserSubmitForm>. The object files UserSubmitFormArticleExtraFieldsItem.pbo and UserSubmitFormPhotoGalleryExtraFieldsItem.pbo are also available for the specific **SubmitType**.

In addition, field specific object files are available at **/Objects/Live-Extrafields/<FieldName>.pbo**

<%ID%>	Contains the extra field ID.
<%Count%>	A forthrunning number.
<%Name%>	Contains the extra field display text.
<%Value%>	Has the value of the extra field. Only available when updating existing data.

UserSubmitFormPhotoGallery.pbo

Works the same way as UserSubmitForm.pbo.

UserSubmitFormPhotoGalleryCategoryList.pbo

Works the same way as UserSubmitFormCategoryList.pbo.

UserSubmitFormPhotoGalleryCategoryListItem.pbo

Works the same way as UserSubmitFormCategoryListItem.pbo.

UserSubmitFormPhotoGalleryErrorFailed.pbo

Works the same way as UserSubmitFormErrorFailed.pbo.

UserSubmitFormPhotoGalleryErrorInvalidInput.pbo

Works the same way as UserSubmitFormErrorInvalidInput.pbo.

UserSubmitFormPhotoGalleryErrorNotCreator.pbo

Works the same way as UserSubmitFormErrorNotCreator.pbo.

UserSubmitFormPhotoGalleryErrorNotLoggedIn.pbo

Works the same way as UserSubmitFormErrorNotLoggedIn.pbo.

UserSubmitFormPhotoGalleryErrorNoUGCAccess.pbo

Works the same way as UserSubmitFormErrorNoUGCAccess.pbo.

UserSubmitFormPhotoGalleryExtraFields.pbo

Works the same way as UserSubmitFormExtraFields.pbo.

UserSubmitFormPhotoGalleryExtraFieldsItem.pbo

Works the same way as UserSubmitFormExtraFieldsItem.pbo.

UserSubmitFormPhotoGalleryPhotoItem.pbo

Object file for formatting the gallery image of the submit form for the user generated by **<pbs:UserSubmitForm SubmitType="ph">**.

<%Counter%>	A forthrunning number.
<%NoOfSections%>	Contains the maximum number of paragraphs available, determined by the **Sections** tag parameter.
<%OrgSections%>	Contains the number of paragraphs on the original story when updating it - otherwise 0.
<%HasContent%>	Contains whether the paragraphs contains an image (=1) or not (=0).

In addition, if updating a gallery the following variables are available:

<%PictureCaption%>	Contains the image title.
<%PictureByline%>	Contains the image byline.
<%PictureSummary%>	Contains the image text.
<%PictureUrl%>	Contains the URL to the image.

UserSubmitFormPhotoGalleryTaxonomyList.pbo

Works the same way as UserSubmitFormTaxonomyList.pbo.

UserSubmitFormPhotoGalleryTaxonomyListItem.pbo

Works the same way as UserSubmitFormTaxonomyListItem.pbo.

UserSubmitFormTaxonomyList.pbo

Object file for formatting the taxonomy words list part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files UserSubmitFormArticleTaxonomyList.pbo, UserSubmitFormPhotoGalleryTaxonomyList.pbo, and UserSubmitFormCCETaxonomyList.pbo are also available for the specific **SubmitType**.

<%Items%>	Contains the taxonomy words formatted using **UserSubmitFormTaxonomyListItem.pbo**.
<%Count%>	Contains the amount of taxonomy words.

UserSubmitFormTaxonomyListItem.pbo

Object file for formatting the taxonomy words part of the submit form for the user generated by <pbs:UserSubmitForm>.The object files of UserSubmitFormArticleTaxonomyListItem.pbo, UserSubmitFormPhotoGalleryTaxonomyListItem.pbo, and UserSubmitFormCCETaxonomyListItem.pbo are also available for the specific **SubmitType**.

<%ID%>	Contains the taxonomy word ID.
<%Count%>	A forthrunning number.
<%Name%>	Contains the taxonomy word display text.
<%Selected%>	Has the value 'selected' if the item is associated with the taxonomy word. Only available when updating existing data.

User Comments

It is now possible to use a new module for user comments instead of the forum module. This module is optimized in regards to performance and is designed only for comments, which makes administration less confusing than when using the Forum module for comments.

> By activating this module, the old Forum module will be disabled.
>
> If you already use the forum, there is need to migrate the old data into the new Comments module. Please contact SAXOTECH Support or Implementation for more information.

User comments are available on stories, photo galleries and CCE elements.

SETUP

[MODULES]

UserComments	[0|1] default 0. Enables the Comments module in the edit interface

[FORUM]

Captcha	[0|1] default 0. Captcha=1 enables captcha validation on user comments
ExternalCaptcha	[0|1] default 0. ExternalCaptcha=1 enables captcha validation from an external source. This ini settings works only if Captcha=1 is enabled.
MaxDaysInactive	Integer, default 180. MaxDaysInactive=100 Controls how many days back the system will look for comments.
NewUserComment	[0|1] default 0. NewUserComment=1 Turns on the usercomments module on the web page
ProtectedPost	[0|1] default 1. ProtectedPost=1 Protects the post against that are not logged in
UserCommentModerated	[0|1] default 0. UserCommentModerated=1 Enables Moderated comment
VerifyUsers	[0|1] default 1. VerifyUsers=1 (must be verified users to post)

Example

The attached zip file, contains all the necessary files to implement the Comments module. This example is based on AJAX.

UserComments_Ajax_Example.zip

Tags

These tags are used with the new Comments module.

- pbs UserComments
  - UserComments.pbo
  - UserCommentsItem.pbo
- pbs UserCommentsSubmit
  - UserCommentsSubmit.pbo
  - UserCommentsSubmitError.pbo

pbs UserComments

<pbs:usercomments>

The UserComments tag works only on story, photo gallery and CCE elements. It is used to display comments for the element.

The tag can also be used to display the latest comments for all stories, photo galleries or CCE objects.

Options

Item	Description

Start	This option makes paginated comments possible. start=N will start the comments on the Nth comment.
AutoUpdate	This option automatically updates the cache file when the cache file's timestamp is older than the last posted comment. This option overrides Cachetime. The accepted values are 0 and 1. 0 is the default. (A ! on the tag is reqired for this to work. `<pbs:!usercomments autoupdate="1">)`
Cachetime	This option controls how often the cache file for this tag should be updated. It is set in minutes and defaults is the general cachetime setting. The value of 0 means the tag is to run uncached.
Category	Filters comments on category. This option only works with sortby=LatestComments, and can only be used for stories and photo-galleries.
Count	This option controls how many messages should be shown. Default value is 10.
Guid From version 6.9.0sr16	Filter comments on the GUID of a content object. This is useful when there are multiple elements on a page with user comments, so the URL parameter cannot be used.
ObjectClass	Controls which object file will be used. Enter objectclass=1 to make the system look for UserComments1.pbo and UserCommentsItem1.pbo. The rules defining which object files will be used are described in the section About object files. You should also read the section on the folders in which the system will look for object files, which can be found in Directory structure > About object files#DirectoryStructure.

SortBy	This option specifies the value by which the list of comments should be sorted. The accepted values are "ModifiedDateTime", "CreatedDateTime", and "LatestComments".
	
	"ModifiedDateTime" is the default. Example:
	
	`<pbs:usercomments sortby=CreatedDateTime>`
	
	"LatestComments" displays latest comments for all stories, photo galleries or CCE objects. It is used in combination with the type option:
	
	`<pbs:usercomments sortby=LatestComments commenttype="AR">`

SortOrder	This option specifies the order in which comments are sorted. The accepted values are desc and asc. Desc is the default. Example: `<pbs:usercomments sortorder=asc>`

CommentType	This option controls the element for which the comments should be displayed. Type works only with sortby=LatestComments.
	
	⚠ Remember to add the CommentType="x" on the pbs:UserCommentsSubmit tag as well.
	
	Type can be any one of the following:
	
	Articles
	`<pbs:usercomments sortby=LatestComments commenttype="AR">`
	
	Photo galleries
	`<pbs:usercomments sortby=LatestComments commenttype="PH">`
	
	CCE
	`<pbs:usercomments sortby=LatestComments commenttype="CCE">`

Object Files

UserComments.pbo

UserCommentsItem.pbo

UserComments.pbo

Variables

<%ArtLink%>	Contains a link back to the article
<%CommentsEnabled%>	Contains a value of "1" if comments for the associated story are disabled.
<%CommentsLocked%>	Contains a value of "1" if comments for the associated story are locked.
<%Items%>	Contains the list of comments.
<%MaxDaysInactive%>	Contains the value defined in publicus.ini Forum MaxDaysInactive=xxx

<%NextLink%>	This variable contains the URL of the next page of comments. `[%{NextLink} <a href="javascript:UserCommentsGetIt('articleComments','<%NextLink%>');">Next</a>%]`
<%ObjectClass%>	Contains the value of the objectclass option in the <pbs:UserComments> tag.
<%PreviousLink%>	This variable contains the URL of the previous page of comments. `[%{PreviousLink} <a href="javascript:UserCommentsGetIt('articleComments','<%PreviousLink%>');">Previous</a>%]`
<%ShortStoryLink%>	Contains a link back to the article (short URL).
<%Sortby%>	Contains the value of sortby tag parameter.
<%SortOrder%>	Contains the value of SortOrdertag parameter.
<%Start%>	Contains the number of the first row.
<%Count%>	Contains the number of rows returned.
<%StoryLink%>	Contains a link to the story associated with the comments.
<%ObjectNetTittel%>	Contains the net title of the story associated with the comments.
<%ObjectSubTittel%>	Contains the subtitle of the story associated with the comments.
<%ObjectSummary%>	Contains the summary of the story associated with the comments.
<%ObjectTittel%>	Contains the title of the story associated with the comments.
<%TotalCount%>	Same as count.
<%TotalNoOfMessages%>	This variable contains the total number of comments for the story.
<%UserCommentsLink%>	Contains a link back to the object comment page.

Some variables are only available when the comments of a specific content item. These are:

<%ItemGUID%> **From version 7.2.0sr30**	The GUID of the item owning the comments.
<%ItemLink%> **From version 7.2.0sr30**	A SEO friendly link to the item owning the comments.
<%SEOLabel%> **From version 7.2.0sr30**	The SEO label of the item owning the comments.

UserCommentsItem.pbo

Variables

<%Author%>	This variable contains information about the user who submitted the comment. Should only be used when not using registred users when posting comments.
<%Body%>	This variable contains the body of the comment.
<%Category%>	This variable contains the category of the story to which the comment was posted.
<%Date%>	Story's publish date
<%CCEGUID%>	This variable contains the CCE GUID if the object is CCE.
<%ISOCreateDateTime%>	This variable contains the creation date and time of the comment in ISO format.
<%ISOModifiedDateTime%>	This variable contains the date and time of the last time the comment was modified in ISO format.
<%ISOPublishedDateTime%>	This variable contains the date and time of the comment's publication in ISO format.
<%ItemCounter%>	This variable contains the place of the current item in the list. The first item has a value of "1".
<%ItemID%>	This variable contains the comment ID in the CMS.
<%Modified%>	This variable contains the date the comment was last modified.
<%PostedByAddress%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the first address line in the location of the user who posted the comment.
<%PostedByAddress2%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the second address line in the location of the user who posted the comment.
<%PostedByAvatarLink%>	This variable contains the URL of the avatar for the user who posted the comment.
<%PostedByCellPhone%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the cell phone number of the user who posted the comment.
<%PostedByCity%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the city location of the user who posted the comment.
<%PostedByCountry%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the countrylocation of the user who posted the comment.
<%PostedByEMail%>	This variable contains the email address of the user who posted the comment.
<%PostedByFirstMessageDate%>	This variable contains the date of the first comment published by the user.
<%PostedByFirstName%>	This variable contains the first name of the user who posted the comment.
<%PostedByGender%>	This variable contains the gender of the user who posted the comment.
<%PostedByID%>	This variable contains the user ID in the CMS
<%PostedByLastMessageDate%>	This variable contains the date of the last comment published by the user.
<%PostedByLastName%>	This variable contains the last name of the user who posted the comment.
<%PostedByNickName%>	This variable contains the nickname of the user who posted the comment.
<%PostedByNoOfcomments%>	This variable contains the number of comment published by the user.
<%PostedByPhone%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the phone number of the user who posted the comment.

<%PostedByPrivatePhone%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the private phone number of the user who posted the comment.
<%PostedByState%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the state location of the user who posted the comment.
<%PostedByUserLevel%>	This variable contains the user level of the user who posted the comment.
<%PostedByZipCode%> **From version 6.6.0sr35, 6.7.0sr7**	This variable contains the zip code location of the user who posted the comment.
<%Published%>	Return 1 if the comment are published.
<%Site%>	This variable contains the two letter site code.
<%StoryId%>	This variable contains the article number of the story to which the comment was posted.
<%Title%>	This variable contains the title of the comment
<%TotalNoOfMessages%>	This variable contains the total number of comments for the story.
<%TotalCount%>	Contains the number of rows returned.
<%UpdateLink%>	This variable contains the URL to modify a comment. A comment can only be modified by the user who posted the comment.
<%DeleteLink%>	This variable contains the URL of the page that will flag the comment to be delete. A comment can only be deleted by the user who posted the comment.
<%YesVoteLink%>	This variable contains the URL necessary to post a Yes vote in the system for the comment. A registered user can only vote Yes one time for the comment.
<%NoVoteLink%>	This variable contains the URL necessary to post a No vote in the system for the comment. A registered user can only vote No one time for the comment.
<%YesVoteCount%>	This variable contains the count of the number of Yes votes for the comment by registered users.
<%NoVoteCount%>	This variable contains the count of the number of No votes for the comment by registered users.
<%ItemGUID%> **From version 7.2.0sr30**	The GUID of the item owning the comment.
<%ItemLink%> **From version 7.2.0sr30**	A SEO friendly link to the item owning the comment.
<%SEOLabel%> **From version 7.2.0sr30**	The SEO label of the item owning the comment.

pbs UserCommentsSubmit

<pbs:usercommentssubmit>

This tag is used to create the form through which users submit comments on stories, photo galleries or CCE objects.

Options

Objectclass	Controls which object file will be used. Enter `Objectclass=1` to make the system look for UserCommentsSubmit1.pbo. The rules defining which object files will be used are described in the section Ab out object files. You should also read the section on the folders in which the system will look for object files, which can be found in Direc tory structure.

CommentType	Defines the type of comment `CommentType=AR|PH|CCE`

Object Files

UserCommentsSubmit.pbo

UserCommentsSubmitError.pbo

UserCommentsSubmit.pbo

Variables

It is possible to use <pbs:captcha> in the object file

If you've enabled captcha in the publicus.ini file in the repository and then if for some reason your captcha isn't working (for instance its not showing up) on the page then users will not be able to submit comments because they have not (and do not have the ability to) complete the captcha verification. The usercommentsubmit form will appear to just post to an empty page.

Item	Description
<%Author%>	This variable is only available when the captcha validation fails. It contains submitted author information. It can be used to refill the Author field in the form when the captcha validation fails.
<%Body%>	This variable is only available when the captcha validation fails. It contains the submitted body text. It can be used to refill the Body field in the form when the captcha validation fails.
<%CommentsLocked%>	This variable contains a value of "1" if comments for the associated Object are disabled.
<%CommentsEnabled%>	This variable contains a value of "1" if comments for the associated Object are enabled.
<%Date%>	This variable contains published date for the associated Object.
<%Category%>	This variable contains category for the associated Object.
<%FromLink%>	This variable is only available when the captcha validation fails. It contains the RedirectURL from the submitted form or the Referer URL.
<%ObjectTitle%>	This variable contains the title of the Object to which the user's comment will be posted.
<%ObjectSummary%>	This variable contains the summary of the Object to which the user's comment will be posted.
<%Postlink%>	This variable contains submit link for the associated Object.
<%Site%>	This variable contains sitecode for the associated Object.
<%StoryId%>	This variable contains StoryId for the associated Object.
<%Title%>	This variable is only available when the captcha validation fails. It contains submitted title info. It can be used to refill the Title field in the form, when the captcha validation fails.
<%ValidCaptcha%>	This variable is only available when the captcha validation fails. Its value is always 0.

All variables from the session file section [USERINFO_x] are available.

<%User_ID%>	This variable contains the user ID like. 1508999
<%User_Email%>	This variable contains the email for the user
<%User_pwd%>	This variable contains the password
<%User_Name%>	This variable contains the both first and last name
<%User_fname%>	This variable contains first name
<%User_lname%>	This variable contains last name
<%User_ip%>	This variable contains the IP adress for the user
<%User_mailformat0%>	This variable contains "selected" if the user have it turned on
<%User_mailinterval0%>	This variable contains "selected" if the user have it turned on
<%User_Avatar%>	This variable contains 0 or 1. 1 = if the user has a avatar
<%User_UserLevel%>	This variable contains, the User level. 5 = can post
<%User_VerifiedStatus%>	This variable contains 0 or 1. 1 = if the user are verified

Form fields

UTF-8 **From version 6.9.0sr10**	Whether the comments are submitted as UTF-8 (=1) or not (=0). Default is 0.

Example

Here is an example of the submit form:

```
[%{ValidCaptcha=0}<font color="red">Incorrect Captcha Code, please enter a new captcha
code.</font>%]

    <form method="post" id="newmessageform" enctype="multipart/form-data"
action="<%postlink%>" name="newmessage" >
        <input type="hidden" name="Published" value="0">
        <input type="hidden" name="UTF-8" value="0">
        <input type="hidden" name="RedirectURL"
value="[%{FromLink}<%FromLink%>%|%Http://google.com%]">
        <br>Title:<br>
        <input type="text"  name="title" maxlength="25" size="25" tabindex="1"
value="<%Title%>">
        <br>Name:<br>
        <input type="text"  name="Author" maxlength="25" size="25" tabindex="2"
value="<%Author%>"> <br>
        <textarea name="Body" type="text" rows="10" cols="35" style="width:560px;"
tabindex="3"><%Body%></textarea><br />
        <pbs:captcha length="3">
        <input id="submitbutton" type="submit"  value="Post a Comment">
    </form>
```

UserCommentsSubmitError.pbo

Variables

<%errorcode%>	If an error occurs upon comment submission, the value of this variable indicates the cause of the error: 1 The user who tried to submit the comment is a read-only user. 2 The user who tried to submit the comment has not verified his email address by clicking the link in the verification email. 3 The user who tried to submit the comment has been banned. 4 The user who tried to submit the comment was not logged in. 5 The user who tried to submit the comment was attempting to update or delete a comment that he did not create. 6 The element to which the comment was to be posted does not exist. 7 The element to which the comment was to be posted is locked. 8 The element to which the comment was to be posted is not enabled for comments.

errorcode

```
[%{errorcode=1}The user who tried to submit the comment is a read-only user.%]
[%{errorcode=2}The user who tried to submit the comment has not verified his email
address by clicking the link in the verification email.%]
[%{errorcode=3}The user who tried to submit the comment has been banned.%]
[%{errorcode=4}The user who tried to submit the comment was not logged in.%]
[%{errorcode=5}The user who tried to submit the comment was attempting to update or
delete a comment that he did not create.%]
[%{errorcode=6}The element to which the comment was to be posted does not exist.%]
[%{errorcode=7}The element to which the comment was to be posted is locked.%]
[%{errorcode=8}The element to which the comment was to be posted is not enabled for
comments.%]
```

Q&A Tags

The functions described in this chapter can be used for both polls and contests. They are set up through the SAXOTECH Online interface. For more information about polls and contests, please see the SAXOTECH Online User Manual.

Please note that you cannot change the number of answers after a poll has accumulated data. Doing this breaks the poll.

- pbs pollarchive
  - PollArchiveItem.pbo
- pbs polldata
  - PollDataResults.pbo
  - PollDataResultsItem.pbo
  - PollDataResultsItemEntry.pbo
  - PollForm.pbo
  - PollFormAlternatives.pbo
  - PollFormAlternativesItem.pbo
  - PollFormItem.pbo
  - Polls_Error.pbo with pbs polldata
  - Polls_Saved.pbo with pbs polldata
  - Poll Statistics
- pbs polllist
  - PollList.pbo
  - PollListItem.pbo
- pbs pollmini
  - Pollmini.pbo
  - PollMiniChart.pbo
  - PollMiniNoPoll.pbo
  - PollMiniQuery.pbo
- pbs pollquestion
- pbs pollsave
  - Polls_Error.pbo with pbs questform
  - Polls_Saved.pbo with pbs questform
- pbs quest_multipage
  - MultiPageQuestIndex.pbo
  - MultiPageQuestPage.pbo
  - MultiPageQuestPageX.pbo
  - MultiPageQuestSubmit.pbo
  - MultiPageQuestWrongAnswer.pbo
  - Multipage self test
  - MultiQuestAltItem.pbo
  - Poll_Error.pbo with pbs quest_multipage
  - Polls_Saved.pbo with pbs quest_multipage
- pbs questform
  - QuestForm.pbo
  - QuestFormAlternativesItem.pbo
  - QuestFormItem.pbo

pbs pollarchive

<pbs:pollarchive>

The poll archive is viewed by accessing a URL like this:

`/apps/pbcs.dll/section?Category=POLLARCHIVE&ShowCategory=OPINION`

In this example, you need to make a template called pollarchive_ovr.pbs. ShowCategory in the URL determines the category from which polls are shown.

Options

Category	The category in which the poll is placed. If this option is not specified, the current category is used.
Chartheight	Determines how high the graph will be in pixels.
Chartwidth	Determines how wide the graph will be in pixels.
Count	Controls how many polls should be shown.
Dateformat	This is how the dates will be presented. This tag uses the regular date format options.

Chartlegend	Determines whether and where a chart legend should be output. The default position is to the right of the graph. The possible options are: Top, Bottom, Left, Right and Off. For example, enter: `chartlegend=Bottom`
ChartType	Specifies the chart type that will be used. By default, the chart type is a bar chart. The following chart types are available: • Pie • Statsbar • Planner • Barstack • Line • Chart • Bar • Areastack • Mline • Tinystat • Tinystatsbar.
ChartValues	Determines whether the system should write the percent and number of votes directly into the graphic. By default, the number of votes is printed next to each bar. The following options are available: off, number, percent and both.
ChartWidth	Specifies the width of the image showing the graph. By default, the system size is used. To make the graph X pixels wide, enter: `ChartWidth=X`
Chartheight	Specifies the height of the image showing the graph. By default, the system size is used. To make the graph X pixels high, enter: `Chartheight=X`
Hor	Used if the graph should be shown horizontal. This can be used with the bar chart type
ShowAxes	Show axes on the graph. Possible values are: both, none
Chart3d	Used for specifying if the chart should be shown in 3d. Possible values are 0, 1. Default is 1
LabelTrim	The length of the label in the poll will be set based on the value of labeltrim. This will be usefull when content of label exceeds chart's width.

PollArchiveItem.pbo

<%ChartLink%>	The link to the chart.
<%EndDate%>	The end date for the poll.
<%Ledetekst%>	The description of the poll.
<%PollID%>	The ID code for the poll. This is used in a hidden field.
<%Question%>	The question that was asked.
<%StartDate%>	The start date for the poll.

<%Votes%>	The total number of votes.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

pbs polldata

<pbs:polldata>

This function is used for a full page poll. To access this poll, you must link to **`/apps/pbcs.dll/poll?Category=Pollcategory`** where Pollcategory is the category in which the poll is placed. It is used on the template **gen_poll.pbs**. After the poll has been posted, the template gen_poll.pbs will be used to show either the thank you message or an error message.

Options

Chartlegend	Determines whether and where a chart legend should be output. The default position is to the right of the graph. The possible options are: Top, Bottom, Left, Right and Off. For example, enter `chartlegend=Bottom` in the tag.
Charttype	Specifies the chart type that will be used. By default, the chart type is a bar chart. The following chart types are available: Pie, Statsbar, Planner, Barstack, Line, Chart, Bar, Areastack, Mline, Tinystat and Tinystatsbar.
Chartvalues	Determines whether the system should write the percent and number of votes directly into the graphic. By default, the number of votes is printed next to each bar. The following options are available: off, number, percent and both.
Imgwidth	Specifies the width of the image showing the graph. By default, the system size is used. Enter `imgwidth=X` to make the graph X pixels wide.
Imgheight	Specifies the height of the image showing the graph. By default, the system size is used. Enter `imgheight=X` to make the graph X pixels high.
Labeltrim	Used to shorten labels, so you can better control the layout. By default, labels are shown as they are entered. Enter `labeltrim=20` to cut off the labels after 20 characters, for example.

Objectclass	Sets the object class. By default, the object class is not set. Enter `objectclass=x` in the tag to set the object class. Replace x with any number between 1 and 999.
Rgbgraphcolors	Defines the colors used for the graph. If you specify three and have five alternatives, the colors will loop, so option four will have the same color as option 1. The system uses the default colors. Enter `rgbgraphcolors=EE3300,33EE00,00EE33` to specify three graph colors.

Showresults	Determines whether web guests will be taken directly to a results page or to a thank you page. The default is to go to the thank you page. Entering
	`showresults=1`
	takes guests to the results page. Entering
	`showresults=0`
	takes guests to the thank you page.

Useobjects	Determines whether object files are used. By default, object files are not used. Enter `useobjects=1` in the tag to use object files.

PollDataResults.pbo

<%Items%>	This field lists all the items.
<%Title%>	The poll title.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

PollDataResultsItem.pbo

<%chart%>	The URL for the chart. If you want to disable the chart's 3D appearance, you must use a regular expression like this: `<%chart$r("(&3d\x3d1)=&3d\x3d0")%>`
<%entries%>	Outputs the entries.
<%PollID%>	Outputs poll id.
<%question%>	Contains the question.
<%totalvotes%>	Shows the total number of votes.

PollDataResultsItemEntry.pbo

<%entryname%>	Shows the name name of the entry.
<%numberofvotes%>	Contains the number of votes.

<%votepercent%>	Contains the percent of votes for the entry - without "%".Here is an example of an PollDataResultItemEntry.pbo file:
	
	`<tr>`
	`<td width="120">`
	`<font face="Verdana, Arial, Helvetica, sans-serif" size="2">`
	`<%entryname%></font></td>`
	`<td width="170">`
	`<img alt="" src="/img/reddot.gif" width="<%votepercent$x(",",".")%>%" height="10"></td>`
	`<td align=right><font face="Verdana, Arial, Helvetica, sans-serif" size="2">`
	`<%votepercent$f("0")%>%</font></td>`
	`</tr>`

PollForm.pbo

If you want to display the results of a poll after submission, the poll form requires a hidden field called 'Redirect' with a value of '/apps/pbcs.dll/pollstats?Kategori=poll&ID=<%id%>'.

<%Action%>	The URL to which the form must be posted.
<%ID%>	The ID code for the poll. This is used in a hidden field.
<%Info%>	The information field from the poll
<%Items%>	Contains all the poll items.

<%MaxAnswer%>	The maximum answer code for the poll. This is used in a hidden field. The Name in this field must be `"max": <input type="hidden" name="max" value="<%MaxAnswer%>">` Please note that the above line is required in the pbo-file.
<%Redirect%>	The redirect URL for the poll. This is used in a hidden field.
<%Title%>	The title of the poll
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

PollFormAlternatives.pbo

<%Items%>	Lists all the alternatives.
<%Name%>	Contains the name.
<%type%>	Contains the poll type.

PollFormAlternativesItem.pbo

<%Alternative%>	Contains the alternative text.
<%Name%>	Contains the question ID.
<%Value%>	Contains the alternative ID.

PollFormItem.pbo

This file is output for every question in the poll.

<%Item%>	Used with the **Required** field to keep track of the question.
<%Question%>	Contains the question.
<%QuestionField%>	Lists all the alternatives.
<%Required%>	Used with the **Item** field to keep track of the question

Polls_Error.pbo with pbs polldata

This page is used if the user does not fill in the name/address form correctly, or does not fill in a required field.

<%backlink%>	Contains the link back to the poll page.

Polls_Saved.pbo with pbs polldata

This page is shown after the poll has been saved.

<%chartlink%>	Contains the link to the results page.

Poll Statistics

The full page poll confirmation page automatically links to this page. It uses the same tags and templates as the full page poll. If you want to link to the page manually, you must use this URL:

`/apps/pbcs.dll/pollstats?Category=POLL&ID=12`

You must specify the ID for the poll you want to show.

pbs polllist

The tag is available from version 7.0.0sr12

The tag <pbs:polllist> can output a list of polls using the following filter parameters and is rendered through the object files of PollList.pbo and PollListItem.pbo.

This makes it possible to create a list of expired polls or contests

ObjectClass	The object class of the object files to use when rendering the tag.
PageSize	The number of items to fetch per page.
PageNumber	The page to fetch.
FromDate	The lower boundary of the active date interval filter.
ToDate	The upper boundary of the active date interval filter.
Categories	A comma separated list of what categories to filter on. The placeholder #category# can be used to specify the current category.
Profiles	A comma separated list of what profiles to filter on. The placeholder #profile# can be used to specify the current profile.
PollTypes	A comma separated list of what poll types to filter on. The following poll types are: Single, Grouped, MultiPageSelfTest, MultiPageContest, MultiPageContestWithRetry, SingleOneUser
Sort	What to sort the list after. The value must consist of a field and direction (asc,desc) for sorting the poll list. [default is asc] The following fields are allowed: categoryname, activefrom, activeto, profilename, title, modified. Default sorting is 'activefrom asc'.

PollList.pbo

The PollList.pbo object file renders a list of polls for the <pbs:polllist> tag.

The variables available are:

<%ItemCount%>	The number of items output.
<%PageNumber%>	The page number being displayed.
<%PageSize%>	The size of each page.
<%TotalCount%>	The total amount of polls found.
<%Items%>	The polls to be listed - each rendered through PollListItem.pbo

PollListItem.pbo

The PollListItem.pbo object file renders a single poll for the <pbs:polllist> tag, and is rendered in the <%Items%> of PollList.pbo

The variables available are:

<%Counter%>	The item's number in the list.
<%ItemCount%>	The number of items presented in the list for the current page.
<%TotalCount%>	The total amount of polls fetched by the tag on all pages.
<%PageNumber%>	The current page number.

<%PageSize%>	The size of each page.
<%IsLastItem%>	Has the value of 1 if it is the last item on the page.
<%ID%>	The ID of the poll
<%PollType%>	The type of the poll.
<%PollTypeName%>	The display name of the poll's type.
<%Title%>	The title of the poll.
<%Description%>	The description field of the poll.
<%Category%>	The category of the poll.
<%CategoryName%>	The display name of the poll's category.
<%Profile%>	The profile of the poll.
<%ProfileName%>	The display name of the poll's profile.
<%ActiveFrom%>	The date from which the poll is active (YYYYMMDD).
<%ActiveTo%>	The date to which the poll is active (YYYYMMDD).
<%ISOActiveFrom%>	The date from which the poll is active (YYYYMMDDHHNNSS).
<%ISOActiveTo%>	The date to which the poll is active (YYYYMMDDHHNNSS).
<%Expired%>	Has the value of 1 if the poll is expired.
<%ObjectClass%>	Contains the value of the objectclass parameter used in the tag.
<%ImageLink%>	Holds the link to the poll's imageif it has one.

pbs pollmini

<pbs:pollmini>

The Mini poll is the most common version of the poll tag. It can be used on all pages. In modern browsers, this poll function allows the user to vote without refreshing the entire page.

The tag can look something like this:

```
<pbs:pollmini category=FRONTPAGE labels="trailing" width="121" height="135"
imgwidth="115" imgheight="60" bgcolor=e9eef9>
```

This tag uses an ilayer for Netscape 4.X and an iframe for Internet Explorer and other browsers that support iframe.

To get this function to work in old browsers, you must place the tag `<pbs:#PollLoad#>` at the bottom of the page right before the `</body>` tag. You can put this tag in your standard footer so it is always included.

Options

Bgcolor	Defines the poll's background color. An iframe is a separate HTML document with its own **<body>** tag. You must write the parameter like this: `bgcolor=#e9eef9`
Category	The category in which the poll is placed. If this option is not specified, the current category is used.

Current	Makes the tag display the poll for the current date, and not the story's date. This option should be present in the tag. Otherwise, web guests will see an old poll when they read an old story.
Height	The height of the Iframe or Ilayer. It should be in the range 100 to 400 pixels, depending on your design.
Imgheight	The height of the result graph shown to people that have voted. It is also shown after the poll is closed.
Imgwidth	The width of the result graph shown to people that have voted. It is also shown after the poll is closed.
Objectclass	Sets object class. By default, this option is not set. Enter `objectclass=x` to set the class where x is any number between 1 and 999.
Pollid	Extracts poll based on the poll id: `<pbs:pollmini pollid="2">`

Profile	The profile in which the poll is placed.
	`<pbs:pollmini profile="1002">`
	Get current profile from the URL.
	`<pbs:pollmini profile="#profile#">`
	will filter on Sports category and on profile 1004.

RGBgraphcolors	Determines which colors the graph bars should have. This option uses the RGB color model. The option is set like this: `rgbgraphcolors="9999cc,cc9999,99cc99"`
Width	The width of the Iframe or Ilayer. It should be in the range 100 to 400 pixels, depending on your design.

This tag uses two object files: One for showing the poll and one for showing the result.

Pollmini.pbo

<%entry1%> to <%entryXX%>	These are the possible entries. By putting [and] at the beginning and end of a line, you indicate that the line should only be output if it contains data.
<%NoPoll%>	Has a value if there is no poll available. It can be used to show a message informing the web guest that no poll is available.
<%posturl%>	The URL to the program that processes the submission.
<%question%>	Outputs the question.
<%systemfields%>	Outputs the hidden system fields that the poll needs to function.

<%Title%>	Outputs the poll title.
<%PollID%>	Outputs the poll ID.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

```
<b><%Question%></b>
<form name="poll" action="<%PostURL%>" method="post">
<%systemfields%>
[%<%Entry1%><br />%]
[%<%Entry2%><br />%]
[%<%Entry3%><br />%]
[%<%Entry4%><br />%]
[%<%Entry5%><br />%]
[%<%Entry6%><br />%]
[%<%Entry7%><br />%]
[%<%Entry8%><br />%]
[%<%Entry9%><br />%]
[%<%Entry10%><br />%]
<br clear="all">
<input type="submit" value="Vote">
</form>
```

PollMiniChart.pbo

<%chart%>	The link to the chart.
<%htmlchart%>	The chart in HTML table format. This option uses the following pbo files to render results: PollDataResults.pbo, PollDataResultsItem.pbo, and PollDataResultsItemEntry.pbo.
<%question%>	Outputs the question.
<%pollExpired%>	Outputs 1 if the poll is expired.
<%votes%>	The number of votes cast.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

PollMiniNoPoll.pbo

Used to output a message if there is no poll available. It is only used when the flag for always using iframe is set in Publicus.ini. The use of this file is automatic. If you want to use the

```
<%NoPoll%>
```

option in Pollmini.pbo, this file cannot be present for the object class.

PollMiniQuery.pbo

If UseIFrameForPollmini in section QA in publicus.ini is off (Default is 1) PollMiniQuery.pbo will be used.

<%PollMiniQuery%>	The URL to the poll. This is also the src link used in iframe polls.
<%ImageLink%> **From version 7.0.0sr12**	Link to poll image
<%PollID%>	Outputs the poll ID.

Jquery example:

AJax Poll

```
<div name="pollMini" id="pollMini<%PollId%>">
</div>
<script>
  $('#pollMini<%PollId%>').load('<%PollMiniQuery%>');
</script>
```

pbs pollquestion

<pbs:pollquestion>

The question poll can be used on all pages. In modern browsers, this poll function allows the user to view the question of active poll.

The tag can look something like this:

```
<pbs:pollquestion category="FRONTPAGE" profile="1002">
```

Options

Category	The category in which the poll is placed. If this option is not specified, the current category is used.
Profile	The profile attached to the poll. If this option is not specified, the current profile is used. For example will this tag filter on current category and the profile of 1002: `<pbs:pollquestion profile="1002">`
PollID **From version 7.0.0sr12**	Use this parameter to force the tag to return information on the poll with the given ID instead of finding one based on the filter options.

pbs pollsave

<pbs:pollsave>

Following tag i used on following templates:

gen_poll.pbs
gen_quest.pbs

This tag is used to save the submitted Q&A, there by it has no options.

Polls_Error.pbo with pbs questform

This page is used if the user does not fill in the name/address form correctly. You can place a JavaScript on the page so guests can go back and complete the form. The script could look like this:

```
<a href="javascript:history.go(-1)">Go back</a>
```

Polls_Saved.pbo with pbs questform

This page contains no variables. It is used to thank the web guest for taking part in the contest.

pbs quest_multipage

<pbs:quest_multipage>

This function uses one page per question instead of having all questions on a single page. To access this contest, you must link to **/apps/pbcs.dll /quest?Category=Questcategory** where Questcategory is the category in which the contest is placed.

This function uses the **gen_quest.pbs** template. This quiz type requires another tag to be put on the gen_quest.pbs template. You should make a template for the category because this function uses a different tag than the normal contest.

This tag gives you a lot of freedom in designing the contest, because it uses object files.

The multipage contest needs the following object files.

MultiPageQuestIndex.pbo

The start page for the contest.

<%Info%>	Outputs the information field for the contest.
<%Link%>	The link to the next page. It should be placed inside a link tag like this: `<a href="<%Link%>">Start</a>`
<%Title%>	Outputs the title field for the contest.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

MultiPageQuestPage.pbo

<%Alternatives%>	This is where the alternatives are output. This tag should be placed inside the form.
<%CurrentAnswers%>	Holds the current answers. It must be present and used like this: `<input type=hidden name="CurrentAnswers" value="<%CurrentAnswers%>">`

Form	The basic form code must include at least the following information.
	`<form name="frmOppgave" action="<%Link%>" method=POST>`
	`<input type=hidden name="CurrentAnswers" value="<%CurrentAnswers%>">`
	`<input type=hidden name="oppg" value="1">`
	`<input type="submit" name="Submit" value="Submit">`
	`</form>`

Freetext answer	It is also possible to have free text answers. To enable this function, use the following code: `<textarea name="answer" value="" cols=50 rows=10></textarea>` This question type is typically used in questions of the type "In twenty words or less, tell us why you should win." You use this code instead of the **<%alternatives%>** field and typically only on one of the questions.
<%Link%>	The link to the next page. It must be placed inside the action parameter of the form.
<%Question%>	Outputs the question.
<%QuestionNumber%>	Outputs the question number.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

MultiPageQuestPageX.pbo

It is possible to design the layout of each question. To do this you must add a number corresponding to the question number after the file name. If you want to make a special design for question number 3, the file must be MultiPageQuestPage3.pbo.

You use the same field names as the regular question page.

MultiPageQuestSubmit.pbo

Form	All input fields must be inside the following form:
	
	`<Form action="<%Link%>" method=post>`
	`Place fields here`
	`</form>`

Hidden fields	The following hidden fields must be present in the form and not changed:
	
	`<input type=hidden name="max" value="<%NumberOfQuestions%>">`
	`<input type=hidden name="quest" value="1">`
	`<input type=hidden name="chart" value="0">`
	`<input type=hidden name="id" value="<%ID%>">`
	`<%Answers%>`

Required fields	Use the following hidden field to specify which fields must be completed. `<input type="hidden" name="requiredfields" value="name,address1,,city,state,zip" >` You can make all fields required.
User information fields	The following fields can be used to get information about the user. `<input type=text name="Title"> <input type=text name="Name"> <input type=text name="Address1"> <input type=text name="Address2"> <input type=text name="City"> <input type=text name="State" size="2"> <input type=text name="Zip" size="5"> <input type=text name="Age"> <input type=text name="Birthdate"> <input type=text name="Email"> <input type=text name="Phone1"> <input type=text name="Phone2"> <input type=text name="Cellphone"> <select name="Sex">` The last field should be a selection box.
<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

MultiPageQuestWrongAnswer.pbo

This version of the contest uses the same object files as the pbs MultipageSelftest. In addition, it has one extra object file. It is used when the user answers a question incorrectly.

This tag uses object files. In addition to the object files used for multipage contest, the following file is used:

<%RestartLink%>	This variable can be used to link back to the beginning of the contest.
<%TryAgainLink%>	This variable can be used to link back to the question that was not answered correctly.

Multipage self test

Multipage self test is a variation of a multipage contest. Instead of the web guest providing his name and number, he will be presented with a page showing how many questions were answered correctly.
The results page uses the object file MultiPageQuestSubmit.pbo to show how many correct answers the web guest had:

MultiPageQuestSubmit.pbo

<%AllCorrect%>	This field has a value if all questions were answered correctly.
<%CorrectAnswers%>	Contains the number of correct answers.
<%NumberOfQuestions%>	Contains the number of questions.

MultiQuestAltItem.pbo

Used to output the alternatives.

<%AltAlfaNumber%>	Outputs a letter corresponding to the alternative number, where A is the first alternative and B is the second.
<%AltNumber%>	Outputs the alternative number. It is used in the input selection like this: `<input type="radio" name="answer" value="<%AltNumber%>">`
<%AltText%>	Outputs the alternative text.

Poll_Error.pbo with pbs quest_multipage

This page is used if the user does not fill in the name/address form correctly. You can place a JavaScript on the page so guests can go back and complete the form. The script could look like this:

```
<a href="javascript:history.go(-1)">Go back</a>
```

Polls_Saved.pbo with pbs quest_multipage

This page contains no variables. It is used to thank the web guest for taking part in the contest.

pbs questform

<pbs:questform>

It is possible to use the Polling component for contests. Depending on the type of contest, different templates will be used. *Single* and *grouped* ref er to options in the editing interface. The tag must be placed on the `gen_quest.pbs` template or a template for a specific category. After the poll has been posted, the template gen_poll.pbs will be used to show either the thank you message or an error message.

To access the contest, you must link to `/apps/pbcs.dll/quest?Category=Questcategory` where Questcategory is the category in which the contest is placed.

To save the submitted content, you need to have <pbs:pollsave> on the responds template. (gen_poll.pbs or gen_quest.pbs)

Options

Category	Determines the category from which contests will be shown.
PollID **From version 7.0.0sr12**	Use this parameter to force the tag to return information on the poll with the given ID instead of finding one based on the filter options.

QuestForm.pbo

This object file is not needed by the system. It is optional. You can use this object file to change the layout of the contest.

In addition to the fields below, you need to have a submit button and the fields for capturing name and address. These fields are described in the section MultiPageQuestSubmit.pbo.

<%Info%>	The information field.
<%questformitems%>	Displays the item file.
<%Title%>	The contest's title.

<%ImageLink%> **From version 7.0.0sr12**	The URL link to the poll's image.

QuestFormAlternativesItem.pbo

Defines the alternatives in the question form. It can only be used with the radio button option.

<%Alternative%>	Displays the alternative text.The fields are typically used like this: `<input name="<%Name%>" type="radio" value="<%Value%>"> <%Alternative%> `
<%Name%>	Used to remember the question number.
<%Value%>	Used to remember the alternative number.

QuestFormItem.pbo

This object file is not needed by the system. It is optional. You can use this object file to change the layout of the contest.

<%alternatives%>	Outputs the alternatives.
<%question%>	Outputs the question.
<%Name%>	The name field.
<%Address1%>	The first address field.
<%Address2%>	The second address field.
<%EMail%>	The email address field.
<%Age%>	The age field.
<%Sex%>	The gender field.
<%Phone1%>	The primary phone number field.
<%Phone2%>	The secondary phone number field.
<%Title%>	The tfitleield.
<%MiddleName%>	The middle name field.
<%Birthday%>	The birthday field.
<%CellPhone%>	The cellphone number field.
<%City%>	The city field.
<%State%>	The state field.
<%Zip%>	The zip code field.
<%US%>	The country field.
<%CustomField1%>	A custom field defined by you.
<%CustomField2%>	A custom field defined by you.
<%CustomField3%>	A custom field defined by you.

Each of the variables corresponds to a poll property field, and will be set to 1 if the field is active on the poll. Also, all of the above variables can be appended `Required` to determine that the field is required. For example,

```
CellPhoneRequired=1
```

if the CellPhone field is required.

Banner Ads

The ad tag is used to pull ads from SAXOTECH OAMS. This system is also used to manage ad codes from DoubleClick and other ad systems.

The ad tag requires interaction with the people selling and managing ads. You must agree on the standard sizes you are going to use for the different ad spots. If you sell ads that are larger than the space left in the templates, the ads will destroy your layout. If you use different sized ads for the same spot, ads will be randomly scaled to the size of one of the ads. This will negatively affect ad appearance.

You can have multiple ad tags on one template and multiple ads for each tag. If you have an ad tag in the header or footer, which is present on all pages, the ad system allows you to display different ads for different categories. Please see the OAMS Chapter to familiarize yourself with how the ad system works.

You enter ads into the system using the SAXOTECH Online interface. For more information, see the OAMS Chapter.

Flash banners
SAXOTECH Online can display Flash banner ads and create rotations of both Flash and animated GIF ads. SAXOTECH Online adheres to industry standards for rich media to ensure properly functioning and supported rich media ads. SAXOTECH Online uses the same solution for proper ad tracking as DoubleClick so third-party ads do not need to be redesigned to be tracked with SAXOTECH Online.

If you wish to serve rich media ads, you must ensure that the ads are built to the requirements specified by Adobe for allowing third-party ad serving solutions to track ads.

In order to track statistics for a Flash ad spot, Flash ads must include a special action script. The clickTAG is the tracking code assigned to an individual ad that allows SAXOTECH Online to register where the ad was displayed when it was clicked. Adding the code to the action script will allow it to navigate the browser to clickTAG by appending the variable data passed in via the OBJECT EMBED tag.

Here is a short guide on how to implement the clicktag feature in different versions of ActionScript. When using the Actionscript 1 or 2 solution, the clickTag parameter is case sensitive. The ActionScript 3 solution is case insensitive.

OAMS outputs the clicktag in this case: clickTAG

Flash 10 (Actionscript 3.0)
ActionScript 3 have changed the way to handle clickTag. Below is an example on how to implement this in flash ads using Actionscript 3. If unsure where to place this code, one solution is to make a clickTag layer in the timeline and handle all the clicktag functions from there.

```
// General functions

// This means that clicktag is not case sensitive
function getClickTag():String {
  for (var key:String in root.loaderInfo.parameters)
    if(key.toLowerCase()=="clicktag")
      return root.loaderInfo.parameters[key];
  return "";
}

// Making an handle means that we do not have to put the logic in each button click
function clickTagEvent(event : MouseEvent) : void {
  openURL(getClickTag(), "_blank");
}

//Make the call to navigateToURL
function openURL(url : String,window : String = "_blank") : void {
  navigateToURL(new URLRequest(url), window);
}

// END General functions

// Button Click - Add for each button, which should open the link
ButtonLabel.addEventListener(MouseEvent.CLICK, clickTagEvent);
```

Flash 8 and 9 (Actionscript 2.0)

```
on (release) {
  if (_root.clickTAG.substr(0,5) == "http:") {
    getURL(_root.clickTAG, "_blank");
  }
}
```

Flash 4, 5, 6, and 7 (ActionScript 2.0)

According to Adobe, 99% of the market have version 8 or newer of the flash plugin installed. If for any reason, something needs to be compiled to an earlier version, please notice that the target for the URL must be _top instead of _blank.
This is important, because _blank in these versions will be seen as a pop-up and bocked as such in some browsers.

```
on (release) {
  if (clickTAG.substr(0,5) == "http:") {
    getURL(clickTAG, "_top");
  }
}
```

Flash 11 (ActionScript 3.0) - Multiple Links

While OAMS cannot handle multiple links in a flash files, these links can now be handled with the help of minor changes of code in the 'fla' file as below. We can only redirect the links in a flash file using this code and we cannot track the particular link which the user has clicked for statistics of individual click. Whenever there is a click on any child link of the flash file the SAXOTECH Online interprets as a click on the banner.

Please find the attached flashExample.fla file for better understanding.

```
//Declaring all the links in an array
var customUri = new Array();

//Master link the one which we enter in Saxotech Online
customUri[0] = "www.saxotech.com/section/CONTACT";

//Child links
customUri[1] = "www.saxotech.com/section/PRODUCTS";
customUri[2] = "www.google.com";
customUri[3] = "www.aspiresystems.com";
customUri[4] = "www.lubera.com";

//Handling the master link with link1
function link1Handler(event : MouseEvent) : void {
  openURL(getClickTag(), "_blank");
}

//Handling all the child links link2, link3, link4 and link5
function link2Handler(event : MouseEvent) : void {
    openURL(linkUrl(1), "_blank");
}

function link3Handler(event : MouseEvent) : void {
 openURL(linkUrl(2), "_blank");
}

function link4Handler(event : MouseEvent) : void {
    openURL(linkUrl(3), "_blank");
}

function link5Handler(event : MouseEvent) : void {
```

```
    openURL(linkUrl(4), "_blank");
}

//Generic function to replace the '/' with '%2F' and creating the respective child
link
function linkUrl(linkNumber:int) : String {
  var urlstr:String = getClickTag();
  var replaceStr:String = customUri[linkNumber].split("/").join("%2F");
  var baseUri:String = customUri[0].split("/").join("%2F");
  var newStr:String = urlstr.split(baseUri).join(replaceStr);
  return newStr;
}

//Generic function to load the master link
function getClickTag():String {
  for (var key:String in LoaderInfo(this.root.loaderInfo).parameters) {
    if (key.toLowerCase()=='clicktag') {
      return LoaderInfo(this.root.loaderInfo).parameters[key];
 }
  }
  return null;
}

//To navigate to the url
function openURL(url : String,window : String = "_blank") : void {
  navigateToURL(new URLRequest(url), window);
}

//To handle the click events of different links
link1.addEventListener(MouseEvent.CLICK, link1Handler);
link2.addEventListener(MouseEvent.CLICK, link2Handler);
link3.addEventListener(MouseEvent.CLICK, link3Handler);
link4.addEventListener(MouseEvent.CLICK, link4Handler);
```

```
link5.addEventListener(MouseEvent.CLICK, link5Handler);
```


- pbs ad
  - AdSpot.pbo
  - FlashBanner.pbo
  - LayerAdSpot.pbo
  - OamsAdSpot.pbo
- SAXOTECH Online ads on other sites

pbs ad

<pbs:ad>

Options

Adlabel	This option only applies for sites using OAMS and is optional. It determines the Ad label to which the ad is assigned. Please see the *SAXOTECH Online User Manual* for more information.By default, the option is not used. The option is used like this: `<pbs:ad spot="TOP1" adlabel="ANCHOR">`
Adsublabel	This option applies only to sites using OAMS and is optional. It determines the Ad sub label to which the ad is assigned. Please see the *SAXOTECH Online User Manual* for more information.By default, the option is not used. The option is used like this: `<pbs:ad spot="TOP1" adlabel="ANCHOR" adsublabel="GATOR">`
Spot	Determines the ad spot for which you want to show ads. This option is mandatory and has no default value.If you put an ad in the system on TOP1, use the parameter `Spot=TOP1` to retrieve the ad. This code must be matched in the ad's management user interface.
SubSite	Determines the subsite from which ads will be pulled when using OAMS. The syntax is: `<pbs:ad spot="SideAd01" Label="FrontPage" SubSite="XY002">` where XY002 is some valid subsite. In case no subsite is supplied, the subsite is read from the publicus.ini.

The system does not require the use of object files. Unless you are trying to do something special, it should not be necessary to use the object files. If you need to use your own object file, it should be saved in the Objects folder root.

If OAMS is enabled, the system will use OAMSAdSpot.pbo and if not it will use AdSpot.pbo and LayerAdSpot.pbo.

The FlashBanner.pbo object file was available in the deprecated PbcsAd.dll module and is not supported by OAMS.dll. Use the AdSpot.pbo, LayerAdSpot.pbo, or OAMSAdSpot.pbo object files, instead. FlashBanner.pbo does not accept objectclass.

AdSpot.pbo

<%adimagelink%>	The image URL.
<%adlink%>	The link for the banner.
<%adlinktarget%>	The target for the link.
<%height%>	The height of the banner.
<%ilayerid%>	The ID for the ILAYER.
<%source%>	The banner's URL.
<%width%>	The width of the banner.

Example code

```
<nolayer>
<iframe src="<%source%>" name="<%ilayerid%>" id="<%ilayerid%>" width="<%width%>"
height="<%height%>" marginwidth="1" marginheight="1" scrolling="no" frameborder="0">
<a href="<%adlink%>" target="<%adlinktarget%>"><img src="<%adimagelink%>" border="0"
height="<%height%>" width="<%width%>" alt=""></a>
</iframe>
</nolayer>
<ILAYER id="<%ilayerid%>" name="<%ilayerid%>" visibility="hide" width="<%width%>"
height="<%height%>">
</ILAYER>
```

FlashBanner.pbo

The FlashBanner.pbo object file was available in the deprecated PbcsAd.dll module and is not supported by OAMS.dll. Use the AdSpot.pbo, LayerAdSpot.pbo, or OamsAdSpot.pbo object files, instead.

<%clickTag%>	The link for the banner.
<%Filename%>	The banner's file name.
<%Height%>	The height of the banner.
<%Width%>	The width of the banner.

Example code

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=5,0,0,0" WIDTH=<%Width%> HEIGHT=<%Height%>>
<PARAM NAME=movie VALUE="<%Filename%>[%?clickTAG=<%clickTAG$j%>%]">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED src="<%Filename%>[%?clickTAG=<%clickTAG$j%>%]" quality=high bgcolor=#FFFFFF
WIDTH=<%width%> HEIGHT=<%Height%> TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/shockwave/download/index.cgi?
P1_Prod_Version=ShockwaveFlash">
</EMBED>
</OBJECT>
```

Location:

```
%AdServer%\templates\pbcsGlobals\flashbanner.pbo
```

LayerAdSpot.pbo

This file contains the same fields as **Adspot.pbo**

OamsAdSpot.pbo

<%Server%>	Link to the OAMS server
<%Site%>	Output the site code
<%Spot%>	Output the spot name
<%Category%>	Output the category
<%Profile%>	Output the profile name
<%AdLabel%>	Output the ad label
<%SubLabel%>	Output the Sub ad label

Example code

```
<script type="text/javascript">
<!--
var rnd = Math.random() + "";
var idn = rnd * 100000000000000000;
document.write('<s'+'cr'+'ip'+'t type'+'="text/javascript"
src="<%SERVER%>/src/<%SITE%>/<%SPOT%>/<%CATEGORY%>/<pbs:!#@PbcsRequestID#>/[%<%PROFILE
%>%|%-1%]/[%<%ADLABEL%>%|%-%]/;idn=' + idn +
';Type=1[%;SL=<%SubLabel%>%]?"><\/script>');
//-->
</script>
```

SAXOTECH Online ads on other sites

The following code is placed at the position where the ad will be shown:

```
<nolayer>
-snip-
</ILAYER>
```

The following code must be placed at the bottom of the page right before the

```
</body>
```

tag:

```
<LAYER>
-snip-
</LAYER>
```

Adspot	The example code displays the ad SKYSCRAPER02. To display the ads for another ad spot you must replace SKYSCRAPER02 with the correct Ad spot code.
Height and width	You must replace all height and width options with their correct values.
Spotname	The code adspot_1 identifies the spot. This code must be unique for each ad on the page. If you put another ad on the page, replace adspot_1 with adspot_2.
URL	You must change the URL to reflect the URL for your site.

Example code

At the ad position

```
<nolayer>
<iframe src="http://sh.us.publicus.com/apps/pbcsad.dll/adsource?
Spot=SKYSCRAPER02,FRONTPAGE&TARGET=_blank&H=240&W=120" name="Foo" id="Foo" width="120"
height="240" marginwidth="1" marginheight="1" scrolling="no" frameborder="0">
<a
href="http://sh.us.publicus.com/apps/pbcsad.dll/adlink?Spot=SKYSCRAPER02,FRONTPAGE&ID=
" target="_blank"><img
src="http://sh.us.publicus.com/apps/pbcsad.dll/adimage?Spot=SKYSCRAPER02,FRONTPAGE"
border="0" height="240" width="120" alt=""></a>
</iframe>
</nolayer>
<ILAYER id="adspot_1" name="adspot_1" visibility="hide" width="120" height="240">
</ILAYER>
```

```
<nolayer>
<iframe src="<%source%>" name="<%ilayerid%>" id="<%ilayerid%>" width="<%width%>"
height="<%height%>" marginwidth="1" marginheight="1" scrolling="no" frameborder="0">
<a href="<%adlink%>" target="<%adlinktarget%>"><img src="<%adimagelink%>" border="0"
height="<%height%>" width="<%width%>" alt=""></a>
</iframe>
</nolayer>
<ILAYER id="<%ilayerid%>" name="<%ilayerid%>" visibility="hide" width="<%width%>"
height="<%height%>">
</ILAYER>
```

At the bottom of the page

```
<LAYER name="adspot_1X"
src="http://sh.us.publicus.com/apps/pbcsad.dll/adsource?Spot=SKYSCRAPER02,FRONTPAGE&TA
RGET=_blank&H=240&W=120" visibility="hide"
onLoad="moveToAbsolute(adspot_1.pageX,adspot_1.pageY);clip.height=240;clip.width=120;
visibility='show';">
</LAYER>
```

Links

The Links functions allow you to build lists of links. You can use these functions to manage all types of links lists on your site.

- pbs links
- pbs linkscategorylist
    - Links.pbo
    - LinksItem.pbo
    - LinksSubcat.pbo
    - Links vote form

pbs links

<pbs:links>

You can also list links from a single category. This function is typically used to include links on a sectionfront or front page. This tag can be used on all templates.

Options

Dump	Dump a logfile of the SQL statement.
SubCategory	Sets the Sub Category ID you want to list. `SubCategory="23955"`
Count	Set the number of items. Default is 5.
MainCategory	Sets the Category ID you want to list. `<pbs:linkscategorylist>`
Objectclass	Set the objectclass.
Region	Sets the Region ID you want to list. `Region="23955"`
SubRegion	Sets the Sub Region ID you want to list. `SubRegion="23958"`
Sort	Sort by "sort field". `Sort="1"`

Random	Determines whether the links should be listed in random order. By default.
	`Random="0"`
	Enter
	`Random="1"`
	to list links in random order.

UseTheme	This must be set if you want to use Theme. Default is 0.
UseKeywords	This must be set if you want to use Keywords. Default is 0.
Theme	Sets the Theme you want to list. `Theme="summer,winter"`
Keywords	Sets the Keywords you want to list. `Keywords="summer,winter"`
MaxQuality	Sets the max quality you want to list. `MaxQuality="2"`
MinQuality	Sets the min. quality you want to list. `MinQuality="3"`
MinAge	Sets the min. age you want to list. `MinAge="3"` 0 = All 1 = Children over 6 2 = Children over 12 3 = Youth 4 = Adults

MaxAge	Sets the max age you want to list. `MaxAge="3"` 0 = All 1 = Children over 6 2 = Children over 12 3 = Youth 4 = Adults

The links tag uses two object files: Links.pbo and LinksItem.pbo.

pbs linkscategorylist

<pbs:linkscategorylist>

This function lists all links in the database. Category headers are included in the list. This tag can be used on all templates.

Options

Count	Controls how many links should be retrieved. The option is set like this: `Count=30` ⚠ When using **Count,** the **Top** and **Last** arguments will be suppressed.
Last	Outputs only the newest link based on your other selections. The option is set like this: `Last=1`
Maincategory	Determines the category from which the links should be retrieved. This number is shown by the category name in the editing interface.
Objectclass	Determines which object class is used.
Region	Determines the region from which the links should be retrieved. This number is shown by the region name in the editing interface.
Sort	By entering `Sort=1` the entries will be sorted by the Sortcode. If the **Last** or **Top** paramet ers have been specified, Sortcode will be the secondary sort field instead of the primary one.
Subcategory	Determines the subcategory from which the links should be retrieved. This number is shown by the subcategory name in the editing interface.
Subregion	Determines the subregion from which the links should be retrieved. This number is shown by the subregion name in the editing interface.

Top	Shows the links with the highest ratings. Entering `Top=10` will result in the 10 highest rated links being shown.
Keyword	Sets the Keyword you want to list. `Keyword="Summer,Winter"`
Usekeywords	Used to show links with the same keywords as the story being viewed. This option can only be used on the article template. By default, this option is not used. Enter `Usekeywords=1` to use keywords for link selection.
Useobjects	Determines whether object files are used. By default, object files are not used. To enable the use of object files, add `Useobjects=1` to the tag.
Theme	Sets the Theme you want to list. `Theme="Summer"`
Usetheme	Used to show links with the same theme as the story being viewed. This option can only be used on the article template. By default, this option is not used. Enter `Usetheme=1` to use the theme for link selection.

The linkscategorylist tag uses two object files: Links.pbo and LinksItem.pbo.

Links.pbo

<%LinkItems%>	Outputs the items.
<%ItemCount%>	Output the total number of items. Will only work with <pbs:links>
<%KeywordsUsed%>	Output the used keyword on the pbs tag. `<pbs:links UseKeywords="1" Keywords="summer">` Will only work with <pbs:links>

<%ThemeUsed%>	Output the used theme on the pbs tag. `<pbs:links UseTheme="1" Theme="summer">` Will only work with <pbs:links>

LinksItem.pbo

This file is output for each link.

<%Counter%>	Countains the counter value.
<%Description%>	The description of the entry.
<%Rating%>	The Rating number
<%Gif_rating%>	A link to the rating image. This rating is given by the staff in the editing interface. The field is used like this: `<img src="<%Gif_rating%>&char=Ç&size=12&Color=0B4770" border="0" alt="" align="middle">` {{&char=«}}determines which character is used for the graphics. This function uses the Wingdings font. `&size=12` sets the size of the image. {{&Color=0B4770}}sets the color of the image.
<%Gif_userrating%>	Provides the URL to the averaged total of user-submitted votes.
<%Imagelink%>	The link to the image. You can add `&Maxw=` to the field to set the image width.
<%Id%>	Provides the ID of the link entry.
<%Logolink%>	The link to the logo. This item must be placed inside an image source tag. You can add `&Maxw=` to the field to set the logo width.
<%Title%>	The title.
<%VoteLink%>	Link to vote action. Read more about Links vote form Remember this must be inside a <form>.
<%URL%>	The URL for entry.

LinksSubcat.pbo

This file is output for each category and subcategory.

<%MainCategory%>	Outputs the name of the main category when it changes.
<%SubCategory%>	Outputs the name of the subcategory when it changes.

Links vote form

To allow web guests to vote on the links, you must set up a form around the list like this:

```
<form action="" method="post" name="Vform" id="Vform">
<input type="hidden" name="vote" value="0">
<input type="hidden" name="redirecturl" value="<pbs:prog>/section?Category=LINKS">
<table width="100%" border="0" cellspacing="0" cellpadding="2">
<pbs:linkscategorylist objectclass=1>
</table>
</form>
```

The form code has the following:

Form setup	`<form action="" method="post" name="Vform" id="Vform">` This line sets up the form. This form has no action, because it is handled by the JavaScript.

Hidden fields	`<input type="hidden" name="vote" value="0">`
	`<input type="hidden" name="redirecturl" value="<pbs:prog>/section?Category=LINKS">`
	These fields are necessary to enable the vote form. The redirect URL should match the page used to view the links.

The following JavaScript must be placed in your template to enable voting:

```
<script language="JavaScript">
function DoVote(p,v) {
       var f = document.Vform;
       f.action = p;
       f.vote.value = v;
       f.submit();
}
</script>
```

In the object file **Linksitem.pbo**, you must enter the following code. This code provides the voting links:

```
<a href="javascript:DoVote('<%Votelink%>',1);">1</a>
<a href="javascript:DoVote('<%Votelink%>',2);">2</a>
<a href="javascript:DoVote('<%Votelink%>',3);">3</a>
<a href="javascript:DoVote('<%Votelink%>',4);">4</a>
<a href="javascript:DoVote('<%Votelink%>',5);">5</a>
```

Page Designer Tags

This section describes the various design and layout tags that can be used.

pbs contentbox

pbs contentbox

> From version 6.9.0

<pbs:contentbox>

The **<pbs:contentbox>** tag is used to display an already defined box from the ContentBox feature.
As the contents of a box is determined by the box itself and its box class, there is no template files available for this tag.

Because of the drag-and-drop functionality, certain restrictions apply to the HTML included in the content boxes and the page in general:

1. All content boxes must have valid HTML. This means that it is not allowed to have unclosed tags.
2. Java script errors on the page that contains the content boxes will disable drag and drop functionality in the edit interface.
3. Document.write is not allowed in HTML elements in content boxes.
4. Z-Index above 998 is not allowed on HTML elements on the page or in the content boxes.
5. Only doctype html 4.01 strict and up is supported at the moment.
6. The following browsers are supported in the edit interface: IE9, Firefox 9 and Safari 5.12
7. The content rendered in the the content boxes must not contain HTML elements with style position: Fixed.

Box General Options

id	The ID of the actual box to display in the tag.
cachetime	The option cachetime is used to determine for how many minutes a contentbox should be cached. By default, the cachetime is the value of the ContentBoxCacheTime publicus.ini setting.

Box Presentation Options

class	Makes it possible to style the div that surrounds the contentbox content by using CSS classes. If more than one class is to be given, the classes need to be delimited by a space.
style	Makes it possible to style the div that surrounds the contentbox content by direct style information. Style information is delimited by semi-colons (;).
fallback **From v. 6.9.0sr4**	Makes it possible to use a macro as fallback if no boxes have been created. It is not possible to specify any macro parameters directly on the content box tag. The value specified in the fallback parameter is the name of the macro. The fallback macro is only cached in the fullpage cache.

Example tags

```
<pbs:contentbox id="sportscenterrail" class="centerbox sportssection">
<pbs:contentbox id="sportsleftrail" class="railbox sportssection" style="float:left">
<pbs:contentbox id="sportsrightrail" class="railbox sportssection"
style="float:right">
```

Net Guest Tags

Net Guest enables you to host online interviews. Web guests can send in questions that are answered through the editing interface. The person answering the questions can be anyplace in the world. The person interviewed only needs access to the Editing Component. You should set up a limited account for the guest, which is deactivated after each session.

- NetGuest.pbo
- pbs netguest
    - NetGuestQuestionItem.pbo
- pbs netguest_list
    - NetGuestList.pbo
    - NetGuestListItem.pbo
- pbs netguest_questionform
    - NetGuestQuestionForm.pbo

NetGuest.pbo

<%Caption%>	The caption for the photo.
<%CaptionHeading%>	The heading for the photo.
<%Heading%>	The heading for the interview.
<%Image%>	This field contains 1 if you have an image attached to the interview.
<%ImgLink%>	The URL for the image.
<%ImgWidth%>	The value of the image the width.
<%ImgHeight%>	The value of the image the height.
<%InfoText%>	The text from the **InfoText** field.
<%TeaserText%>	Outpur teaser field.
<%Photo%>	The photo credit.
<%Questions%>	This field displays the questions.
<%QuestionEditLink%>	A the link to the Question form.

pbs netguest

<pbs:netguest>

The template is called **gen_netguest.pbs** and is accessed through the address **/apps/pbcs.dll/netguest**.

Options

Count	Controls how many questions are shown. By default, all questions are shown. Enter `count=#` to show a specific number of questions.
Enddate	Sets the end date for the list of questions. By default, all questions are shown. Enter `enddate=YYYYMMDD` to show only questions entered before this date.
ObjectClass	Used for template lookup, so different object files can be used to generate the output.

Pictureheight	Sets the maximum height of the image.
Picturewidth	Sets the maximum width of the image
Sort	Sets the sort order of the answered questions. By default, the answers are output in the order in which they were answered. Enter `sort=written` to output the answers in the order in which the questions were written.

Sortdirection	Controls the sorting of the questions and answers. By default, the questions and answers are sorted ascending. Enter
	`sortdirection=descending`
	to sort the questions in descending order, with the newest question first. Enter
	`sortdirection=auto`
	if the sort should be descending when the interview is active and ascending when its not.

Start	Sets the start number for the first question. By default the first question is shown. Enter `start=#` and the system will skip the first #-1 questions.
Startdate	Sets the start date for the list of questions. By default, all questions are shown. Enter `startdate=YYYYMMDD` to show only questions entered after this date.

NetGuestQuestionItem.pbo

<%\Author%>	The name of the question submitter.
<%Answer%>	This field contains the answer.
<%Birthday%>	The submitter's birthday.
<%Custom1%>	The information from the first custom field.
<%Custom2%>	The information from the second custom field.
<%\Email%>	The submitter's email.
<%Heading%>	The title for the question.
<%Phone%>	The submitter's phone number.
<%Question%>	The question.

pbs netguest_list

<pbs:netguest_list>

This function is used to list future and past guests. This tag can be placed on all templates. In the example, it is placed on the template **gen_netguest.pbs**.

Options

Archive	Makes the tag list past guests instead of future guests.
Category	Determines the category for which guests will be shown. By default, guests for all categories are shown. To show only guests for the current category enter `category=#category#` This function can also be used to show guests from the current category and subcategory.
Count	Determines how many guests the tag should list. By default, this is 1. To list more guests enter `count=#` where # is a number between 0 and 999. 0 will list all guests. 1 through 999 will list the selected number of guests.
Objectclass	Sets the object class for the object files.
Showref	Controls how many characters of the teaser is shown. By default, the entire teaser is shown.To change the amount of characters from the teaser that is shown enter `showref=#` where # is a number between 1 and 9999.

NetGuestList.pbo

<%Items%>	Outputs the items.

NetGuestListItem.pbo

<%Active%>	This field has the value 1 if the questionlink is active on the guest.
<%\Date%>	Contains the date.
<%Heading%>	Contains the heading.
<%Img%>	This field has the value 1 if there is an image attached to the meeting.
<%ImgLink%>	The link to the image.
<%InfoText%>	The text from the teaser field in the editing interface.
<%IsoDate%>	The date in ISO format.

<%Link%>	The main page.

pbs netguest_questionform

<pbs:netguest_questionform>

The Question form is linked to from the main guest page. The template is called **gen_ngqedit.pbs**.

NetGuestQuestionForm.pbo

This line is used to set up up the form. The action for the form must be

```
<%SaveAction%>
<form action="<%SaveAction%>" method="post">
```

Fields

Birthday	`<input type="text" name="Birthday" size=30>` Captures the birthday.
Custom field 1	`<input type="text" name="Custom1" size=30>` Captures the first custom field. This field can be used for whatever extra information you want to capture.
Custom field 2	`<input type="text" name="Custom2" size=30>` Captures the second custom field. This field can be used for whatever extra information you want to capture.
E-mail address	`<input type="text" name="Email" size=30>` Captures the e-mail address.
Name	`<input type="text" name="Author" size=30>` Captures the name of the user.

Phone	`<input type="text" name="Phone" size=30>` Captures the phone number.
Question	`<textarea name="Question" cols=49 rows=5></textarea>` This field is used for the question.
Title	`<input type="text" name="Heading" size=30>` This field is used for the title.

This hidden field is used by the system to manage the questions:

```
<input type="hidden" name="GID" value="<%GID%>">
```

This field sets the page to which the web guest is sent after submitting the question. You can make this any page. The field `<%NetGuestLink%>` contains the link back to the main Net Guest page:

```
<input type="hidden" name="ReturnPath" value="<%NetGuestLink%>">
```

Subscriptions

Subscriptions can be set on all categories. When subscriptions are enabled, you need to have the template gen_login.pbs to handle logins. Subscriptions are a separate function from registration.

User information is stored in the SAXOTECH Online database. The user management screen is documented in the Subscriptions chapter in the SAXOTECH Online User Manual. This user database is not related to the White Pages database. This is a separate database, which is only used for protected pages.

Instructions for setting up a category for subscription can be found in the Subscriptions chapter in the SAXOTECH Online User Manual.

- Login page
  - pbs login
  - pbs logininfo
  - pbs target
- pbs changepassword
- pbs global
- pbs subscriberdata
  - SubscriberData.pbo
- pbs userid
- Self-created users

Login page

Templates

The default template for login is **gen_login.pbs**. If you would like to have a specific template for a category, replace **gen** with the category ID. When a web guest tries to access a section that is protected, he will be directed to this page.

SAXOTECH Online needs the following tags and fields in the template.

Fields

PBS_UserID	Passes the user name to the system.
PBS_Password	Passes the password to the system.
Usefailedpage	Enables the failed page function. Normally, the web guest is returned to the login page if login fails. If this field is present with the value 1, the template gen_loginfailed.pbs will be shown instead. The field is typically used like this: `<input type=hidden name="usefailedpage" value=1>`

Example

```
<form action="<pbs:login>" method="post" name="Login" id="Login">
<input type="hidden" name="PBS_Target" value="<pbs:target>">
Username: <input type="text" name="PBS_UserID" size="14" class="input"><br>
Password: <input type="password" name="PBS_Password" size="14" class="input"><br>
<input type="submit" value="Continue &gt;&gt;" class="button">
</form>
```

pbs login

<pbs:login>

This tag must be used as an action by the template:

```
<form action="<pbs:login>" method="post" name="Login" id="Login">
```

pbs logininfo

<pbs:logininfo>

Outputs the information field from the subscription category.

pbs target

<pbs:target>

⚠ This tag does not work with Frontend cache or LazyWrite of sessions.

This tag simply outputs the 'target' session file value for the customer. Specifically, it outputs the session field of:

```
[DATA]
<abontype>_Target=X
Target=Y
```

trying to output the value of X before falling back to returning Y.

It takes no parameters and has no templates.

This tag is placed inside a hidden field called PBS_target:

```
<input TYPE=hidden NAME="PBS_Target"  value="<pbs:target>">
```

The system uses this field to return the web guest to the correct sectionfront or story.

pbs changepassword

<pbs:changepassword>

This tag outputs the URL to which the form must be submitted. This is also the URL for the change password form.

The user must be logged in to change her password. To access the change password page, you must use this URL:

```
/apps/pbcs.dll/nyttpassord
```

You can specify a category.

Templates

This function uses the template **gen_nyttpass.pbs**.

Form

The form is set up with the following code:

```
<FORM name="frmChangePassword" method="post" action="<pbs:changepassword>">
```

Fields

Old password	The user enters the old password in this field: `<input type="password" name="PBS_OldPassword" size=12 value="">`
New password	The user enters the new password in this field: `<input type="password" name="PBS_NewPassword1" size=12 value="">`
Repeat new password	The user re-enters the new password in this field: `<input type="password" name="PBS_NewPassword2" size=12 value="">`
Target	(Hidden field) Sets the page to which the user is sent when the password is changed: `<input type="hidden" name="PBS_Target" value="/apps/pbcs.dll/frontpage">`
User ID	(Hidden field) Stores the user ID: `<input type="hidden" name="PBS_UserID" value="<pbs:userid>">`

Submission form

You should use the following code to check that the passwords the user has entered are not blank and that the new password is the same in both fields:

```
<INPUT type="button" value="Continue" onClick="javascript:CheckNewPass();">
<script language="javascript">
function CheckNewPass () {
var submitForm = true;
var NewPass1 = document.frmChangePassword.PBS_NewPassword1.value;
var NewPass2 = document.frmChangePassword.PBS_NewPassword2.value;
if (NewPass1 == "" || NewPass1 == null || NewPass2 == "" || NewPass2 == null) {
submitForm = false;
alert("You must enter your new password twice to confirm that your password will be
changed.");
}
if (submitForm) {document.frmChangePassword.submit();}
}
</script>
```

pbs global

<pbs:#global#>

You can also display the content of the <%@loginfailedmessage%> variable using the <pbs:#global#> tag like this:

```
<pbs:!#global# object="<%@loginfailedmessage%>">
```

This tag should be run uncached because the message can vary from user to user.

Templates

This template is used to redesign the page to which the web guest is sent when login fails and the **Usefailedpage** option is set. The template is called **gen_loginfailed.pbs**.

This page has no special tags. You can use all the tags from pbs subscriberdata. It has a special global variable that can be used in all object files used on the page.

The variable is <%@loginfailedmessage%>.

pbs subscriberdata

<pbs:subscriberdata>

This function allows you to show information about a logged-in web guest. The tags cannot show information from external subscription systems like Clickshare.

Templates

This tag can be used on all templates.

Options

Field	This section is used to specify which field should be available. Only one field can be specified.

Fields	This section is used to specify which field should be available. Multiple fields can be specified. The available fields are: • Userid - The web guest's ID. This is the ID that he uses to log on to the site. • Username - The web guest's name. • Email - The web guest's email address. • Phone - The web guest's phone number. • Fax - The web guest's fax number. • Description - A description of the web guest. • Fromdate - The date from which the account is active. • Todate - The expiration date for the account. • Password - The web guest's password. • Changepassword - This field has the value 1 if the change password flag is set. • Loggedin - This field has the value 1 if the user is logged in. This field is only available when you use the **Type** option in the tag.
Type	This option is used to specify the subscription type. This option does not have a default value. Specify the category ID for the subscription type. If you use Clickshare, the type *Clickshare* is also available.

SubscriberData.pbo

The fields you have specified in the **Field** or **Fields** sections will be available here as variables.

pbs userid

<pbs:userid>

This tag outputs the user ID for the logged in user.

Self-created users

The self-creation option lets the users create their own accounts. To use this option, you must create a template as described in the following section.

Users can be added to the database in several ways.

If you have an external database of user names and passwords, SAXOTECH Online can import these names and passwords into the database, if your system can export them. This option requires integration with your system and is done on a case-by-case basis. Please contact SAXOTECH if you want to use this option.

The SAXOTECH Online interface also allows your staff to enter users directly into the database. This is described in the *SAXOTECH Online User Manual.*

Templates

When using the self creation option you must create a template in which the users can create their accounts. This file can be named whatever you like, but SAXOTECH recommends **adduser_ovr.pbs** so it can be accessed as a sectionfront.

Fields

The following fields are mandatory. Omitting any of this code will break the page.

Date range	Set the date range in which the account is active. These dates are entered in the YYYYMMDD format.
	`<input type="hidden" name="Fra_dato" value="00000000">`
	`<input type="hidden" name="Til_dato" value="99999999">`

Email address	The user's email address. `<input type="text" name="Email" value="">`
ErrorURL	The URL to which the user will be sent if the form is filled-in incorrectly. `<input type="hidden" name="ErrorURL" value="/apps/pbcs.dll/misc?URL=/misc/error.pbs">`
Fax	The user's fax number. This field is optional. `<input type="text" name="Fax" value="">`
Form code	The template must start with the following line: `<form name="frmEditAbo" method="POST" onSubmit="return SjekkData(this)" action="/apps/links.dll/savelink?Type=6">` This code must not be changed. Changing it will break the page.
Javascript	The JavaScript in the top of the example code must be present.
Name	The user's name. `<input type="text" name="Brukernavn" value="">`
Newspaper code	Used to set the newspaper code. It must be set to your code. `<input type="hidden" name="Avis" value="GE">`

Other fields	The following fields must be present:
	
	`<input type="hidden" name="OrgBrukerid" value="">`
	`<input type="hidden" name="Byttpassord" value="0">`
	`<input type="hidden" name="CheckMode" value="0">`
	`<input type="hidden" name="win" value="0">`
	`<input type="hidden" name="nye" value="">`

Password	The user's password. `<input type="password" name="Password" value="">`
Phone	The user's phone number. This field is optional. `<input type="text" name="Tlf" value="">`
RedirectURL	The URL to which the user will be sent if the form has been filled-in correctly. `<input type="hidden" name="RedirectURL" value="/apps/pbcs.dll/section?category=obituaries">`

Subscribtion type	Set the category to which the subscribers should be added. The value of these fields must match the category you have created through the editing interface.
	
	`<input type="hidden" name="OrgAboType" value="SUBSCRIBE">`
	`<input type="hidden" name="Abon_type" value="SUBSCRIBE">`

User ID	The user name with which the user logs on to your site. `<input type="text" name="Brukerid" value="">`

User Registration

User login can use the information in the White Pages to allow only registered users to see content or add events to the events calendar. User login is a result of user registration. Users can register to have their names in the e-mail catalog. The same database that is used for the White Pages is also used for user verification when users submit events or sign up for a newsletter.

- Setup user registration
- Advanced login
    - pbs ureg_email
    - pbs ureg_message
- Edit user registration
- Log out
- Lost password
    - PwdSent.pbo
- pbs adregusertargets
- pbs extra_income
- pbs reguserinfo
    - Userreginfo.pbo

Setup user registration

To have a section requiring user registration, you must set the **Registration** option to either `Simple` or `Advanced` for the category in the category register. Setting the option to `Simple` will present the user with the template **gen_userreg.pbs** before the page is shown. Setting the option to `Advanced` will present the user with the template **gen_userreglogin.pbs**.

Please see Advanced login for information about using the Advanced option for presenting a login page.

Script

The following JavaScript will open a dialog box if the user says her age is below 13.

```
<script type="text/javascript">
function Checkage(theForm) {
if (theForm.age[0].selected==true) {
 alert('COPA prohibits user registration by children below the age of 13.');
 document.location.href = "/under13";
 return false;
}
}
</script>
```

Form

The form code must be present for the form to work.

```
<form method="post" name="information2" action="<pbs:reguser>" onSubmit="return
Checkage(this);">
```

Fields

The only fields that are required are: Email address, first name and last name. Password and repeat password are only necessary when the generate password function is not used.

It is possible to add custom fields to your users. These fields are not available through the edit interface but they are available on the templates. See the options for the pbs reguserinfo tag.

The fields are added to the template as follows:

<input type=text name="extra_age" value=<pbs:#extra_age#>>

In the example above, a new field called age will be stored with the information written in the input form. The extra field's name can be used for target advertising in OAMS. So, if you create target advertising, then the names of the fields in OAMS and the extra fields from the user registration will make target advertising work.

Address 1	The first address line. `<input type="text" name="address" size="15" value="<pbs:#ureg_address#>">`
Address 2	The second address line. `<input type="text" name="address2" size="15" value="<pbs:#ureg_address2#>">`
Age	Verifies that the user is over the age of 13. This is done to comply with the United States Child Online Protection Act. `<select name="age" SIZE=1> <option value="13 or under" selected>under 13</option> <option value="13over">over 13</option> </select>`
Cellphone	The cell phone number. `<input type="text" name="cellphone" size="12" value="<pbs:#ureg_cellphone#>">`
Cookiename	(Hidden field) Sets the cookie name that is used to store the username and password `<input type="hidden" name="cookiename" value="usernamepassword">`
City	The city. `<input type="text" name="city" size="15" value="<pbs:#ureg_city#>">`
Date of birth	The user's date of birth. `<input type="text" name="dob" size="8" value="<pbs:#ureg_dob#>">`

Email format	Selects the email format.
	
	`<select name="mailformat">`
	`<option value="0">Select format >>`
	`<option value="1"`
	`<pbs:#ureg_mailformat1#>>Text`
	`<option value="2"`
	`<pbs:#ureg_mailformat2#>>HTML`
	`</select>`

Email address	The user's email address. It is used as user name. `<input type="Text" name="email" value="<pbs:#ureg_email#>" size=30>`
Email subject	(Hidden field) Sets the email subject. `<input type="hidden" name="emailsubject" value="User registration">`

Email text	Creates the email body. The field can have multiple lines. All fields in the registration form can be used here.
	
	`<input type="hidden" name="emailtext" value="You are now registered at widgets.com with the following information:`
	`Email: #ureg_email#`
	`Password: #ureg_pwd#`
	`Other fields go here...">`

First name	The user's first name. `<input type="Text" name="fname" value="<pbs:#ureg_fname#>" size="30" maxlength="40">`

Gender	This field allows the user to pick a gender.
	
	`<select name="sex">`
	`<option value="M"`
	`<pbs:#ureg_sex_male#>> Male`
	`<option value="F"`
	`<pbs:#ureg_sex_female#>> Female`
	`</select>`

Generate password	Use this hidden field if the system should generate a password for the user. When this option is used, the fields enabling the user choose a password must not be present. The email option must be enabled so that the password can be mailed to the user.
	
	`<input type="hidden"`
	`name="generatepassword" value="1">`

Home phone	The home phone number.
	
	`<input type="text" name="homephone"`
	`size="12"`
	`value="<pbs:#ureg_homephone#>">`

Homepage	The user's homepage.
	
	`<input type="text" name="homepage"`
	`size="15"`
	`value="<pbs:#ureg_homepage#>">`

ID	(Hidden field) This field must also be present to keep track of the ID code.
	
	`<input type="hidden" name="id"`
	`value="<pbs:#ureg_id#>">`

Last name	The user's last name.
	
	`<input type="Text" name="lname"`
	`value="<pbs:#ureg_lname#>" size="30"`
	`maxlength="40">`

Middle name/Initial	The user's middle name or initial. `<input type="Text" name="mname" value="<pbs:#ureg_mname#>" size="30" maxlength="40">`
Password	The user's password. `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size=20>`
Phone number	The user's phone number. `<input type="Text" name="phone" value="<pbs:#ureg_phone#>" size="15" maxlength="40">`
Redirect URL	(Hidden field) Sets the redirect URL. This field should only be used when the server generates passwords. It directs the user to the page informing her that a password has been sent to her e-mail address. `<input type="hidden" name="RedirectURL" value="/">`
Region	(Hidden field) This field must be present. `<input type="hidden" name="Region" value="0000">`
Repeat password	This field is used to repeat the password. This is done so the user can verify that the password is typed correctly. `<input type="Password" name="pwdr" value="<pbs:#ureg_pwdr#>" size="20" maxlength="20">`
Send email	(Hidden field) Sends an email with the user information. `<input type="hidden" name="sendemail" value="1">`

Sender's email	(Hidden field) Sets the sender's email address. `<input type="hidden" name="fromaddress" value="someaddress@example.com">`
Sender name	(Hidden field) Sets the sender's name. `<input type="hidden" name="fromname" value="Example Name">`
Show in directory	This checkbox allows the user to determine whether she will be shown in the White Pages. `<input type="checkbox" name="show" <pbs:#ureg_show#> value="1">`
State/Country	The country or state. `<input type="text" name="country" size="15" value="<pbs:#ureg_country#>">`
Submit	This button submits the information to the server. `<input type="Submit" name="Submit" value=" Register ">`
Tip	This field is used for a newsletter. The newsletter is unavailable at the moment. `<input type="checkbox" name="tip" <pbs:#ureg_tip#>>`

Tip interval	Sets the tip interval for the newsletter. This a separate newsletter function separate from the headlines newsletter. This function is currently not available.
	
	`<select name="mailinterval">`
	`<option value="0">Select interval >>`
	`<option value="1"`
	`<pbs:#ureg_mailinterval1#>>Daily`
	`<option value="7"`
	`<pbs:#ureg_mailinterval7#>>Weekly`
	`</select>`

Use cookie	This checkbox allows the user to store the user name and password in a cookie so that she does not have to type the user name and password each time a protected section is accessed.
	
	`<input type="checkbox"`
	`name="usecookie">`

White pages	This checkbox controls whether the user is willing to have her name in your email catalog.
	
	`<input type="Checkbox" name="show"`
	`value="1" <pbs:#ureg_show#>>`

Zipcode	The zip code
	
	`<input type="text" name="zip"`
	`size="5" value="<pbs:#ureg_zip#>"">`

Data retrieval fields

These tags are used to pass information back to the form if the user fills in a field incorrectly must complete the form again. These fields save the user from retyping information.

There is a matching tag for each field in the form. For example:

To retrieve the information for the field email, use the tag:

```
<pbs:#ureg_email#>
```

This field is entered in the value field for the input field with the name email like this:

```
<input type="Text" name="email" value="<pbs:#ureg_email#>" size="30" maxlength="40">
```

Advanced login

The information in the White Pages can be used to allow only registered users to see content or add events to the events calendar.

To add login to a section, you must set the **Registration** option to **Advanced** on the category in the category register. The user will be presented

with the login form before the she can see the page.

To force an already-logged-in user to log in again, you can use the option **&ForceLogin=1** in the link to the page.

> From version 6.4.0:
> To enable Facebook single-signon you must add the option **&FBLogin=1** in the link to the page. Default is 0.

Template

The template used for the login form is called **gen_userreglogin.pbs**.

Form

The following code is used to set up the form:

```
<form METHOD="POST" name="information" action="<pbs:reguser>"
onSubmit="pbsSetCookie(this)">
```

Fields

Cookiename	Use this hidden field to set the name for the cookie that stores the username and password. `<input type="hidden" name="cookiename" value="usernamepassword">`
Editing info	Use the following to create a link to the page for editing the user information: `<a href="frontpage?ForceUserreg=1&Edit=1">Edit your info</a> `

Forgotten password	You can also link to a page where the user can email her password to herself.
	
	`<a href="javascript:NewWindow(450,300,'/apps/pbcs.dll/misc?URL=/misc/password_email.pbs')"> Forgot your password? </a>`
	
	or
	
	`<a href="/apps/pbcs.dll/misc?URL=/misc/password_email.pbs"> Forgot your password? </a>`
	
	Which link you use depends on whether the page is a pop-up or not.

Link for registering	Use the following to create a link to the user registration page. `<a href="frontpage?ForceUserreg=1">Register yourself</a> `
Login	This hidden field must be in the form to tell the system that the user should log in, and not register a new user. The field is used like this: `<input type="hidden" name="Login" value="1">`
Password	The password field is called **pwd.** `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size="20" maxlength="20">`
RedirectURL	Use this hidden field to send the user to a page other than the one for which you have set user login. You must add the following hidden field to the form: `<input name="RedirectURL" type="hidden" value="someURL">?`

Usecookie	You can use the field **usecookie** to allow the user to store her user name and password in a cookie so that she do not have to log in each time. `<input type="checkbox" name="usecookie" value="1">`
Username	The username field is called **e-mail** because the e-mail address is used as the username. `<input type="Text" name="email" value="<pbs:#ureg_email#>" size="20" maxlength="40">`

Script

The following JavaScript needs to be called:

```
<script language="javascript" src="/apps/cookiesearchparams.js"></script>
```

This script contains all the cookie functions used by the form.

pbs ureg_email

<pbs:#ureg_email#>

Outputs the user's email address if she is already logged in.

pbs ureg_message

<pbs:#ureg_message#>

The system uses the function

```
<pbs:#ureg_message#>
```

to output messages from the system when the user is registering.

By default, a value of "Wrong Password." indicates that the user has entered an incorrect "email/username" or "password". Depending on the hidden field present in the HTML form, this message can be altered by use of object files.

```
<pbs:#ureg_message# object="[%<div><%variable$r(\"Wrong=Incorrect\")%></div>%]">
```

The following field values, if present in the form's hidden fields, will determine which object file will be read in and displayed.

Parameter: edit=1
Object File: AuthMessageUserRegEdit.pbo

Parameter: login=1
Object File: AuthMessageUserRegLogin.pbo

Parameter: [No Parameter]
Object File: AuthMessageUserReg.pbo

e.g.

```
<form METHOD="POST" name="login" action="/apps/pbcs.dll/reguser?Category=&All=1">
<input type="hidden" name="Login" value="1" /> <!-- HIDDEN "LOGIN" FIELD WILL USE
'AUTHMESSAGEUSERREGLOGIN.PBO' -->
<input type="hidden" name="cookiename" value="usernamepassword" />
<table>
  <tr>
    <td colspan="2"><h2>Log in here:</h2></td>
  </tr>
  <tr>
    <td><strong>E-mail Address:</strong></td>
    <td><input type="text" name="email" value="" size="45" maxlength="75" /></td>
  </tr>
  <tr>
    <td><strong>Password:</strong></td>
    <td><input type="Password" name="pwd" value="" size="45" maxlength="75" /></td>
  </tr>
  <tr>
    <td colspan=2><input type="submit" value="Log In" /></td></tr>
</table>
</form>
```

Edit user registration

This function allows the user to change White Pages information.

This tag is an example of how to output custom user information from the system. In this case, the information is user income. These tags will start with "extra_" followed by the field name of the desired information. This tag is used on the Edit Form.

To let the user update her information, you must add `&ForceUserReg=1&Edit=1` to the link that takes her to the White Pages page.

The link should look like this:

```
<a href="frontpage?1&ForceUserreg=1&Edit=1">Edit your info</a><br>
```

Templates

The logon form uses the template **gen_userregedit.pbs**.

The edit template is called **gen_userregeditform.pbs**. The input fields are the same as used in the registration form.

Logon form

This line sets up the form.

```
<form METHOD="POST" action="<pbs:reguser>">
```

Fields

Edit	(Hidden field) Sets the next page to be the edit form. `<input type="hidden" name="edit" value="1">`
Password field	Captures the password. `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size="20" maxlength="20">`

Edit form

After the user has logged on to the system, the form for editing her current information is shown. The form is built with the following fields.

This line sets up the form.

```
<form METHOD="POST" name="information2" action="<pbs:reguser>">
```

Fields

These fields must be present for the form to work:

<input type="hidden" name="Region" value="0000">
<input type="hidden" name="edit" value="1">
<input type="hidden" name="id" value="<pbs:#ureg_id#>">
<input type="hidden" name="email" value="<pbs:#ureg_email#>">

⚠ The e-mail adresse can not be edited through the frontend, the current email must be a hidden field on the form.

Log out

To log out, you must use the following JavaScript to delete the cookie:

```
<script Language="Javascript">InitCookieSearchParams(document.login)</script>
<a href="javascript:EatCookie();">Log out</a><br>
<script>
function EatCookie() {
document.cookie="usernamepassword=;Path=/"
document.cookie="UserRegID=;Path=/"
document.location="/apps/pbcs.dll/frontpage?RegLogout=1"
}
</script>
```

OAMS Ad targeting eat cookie

To use the eat cookie function with OAMS ad targeting, you must add a line to the JavaScript:

```
<script Language="Javascript">
InitCookieSearchParams(document.login)
</script>
<a href="javascript:EatCookie();" class="white">Log out</a>
  |  
<a class="white" href="/apps/pbcs.dll/section?Category=REGISTER">Edit your
settings</a>
<script>
function EatCookie() {
document.cookie="usernamepassword=;Path=/"
document.cookie="UserRegID=;Path=/"
document.cookie="PBCSSESSIONID=;Path=/"
document.cookie="OAMS_TARGETS_LIST=;Path=/"
document.location="/apps/pbcs.dll/frontpage?RegLogout=1"
}
</script>
```

This will ensure that the web guest is served non-targeted ads upon log out.

Lost password

This function lets the user receive her password through e-mail if it has been forgotten.

The link can look like this:

```
<a href="/apps/pbcs.dll/misc?URL=/misc/password_email.pbs">
```

If you want the form to open in a pop-up window, use this link:

```
<a href="javascript:NewWindow(450,300,
'/apps/pbcs.dll/misc?URL=/misc/password_email.pbs')">
```

Form

You set up the form with the following line:

```
<form name="emailform" action="<pbs:prog>/tips?sp=1" method="post">
```

Fields

Email address	The **To** field is used for the email address to which the password should be sent. `<input type="text" name="to" size="25">`

Email subject	Used to give the e-mail a subject line. `<input type="hidden" name="title" value="Password reminder">`
FailedNext	(Hidden field) Used to set the page to which the user will be taken if the email address is not in the database. `<input type="hidden" value="<pbs:prog>/misc?URL=/misc/password_emailfailed.pbs" name="failedNext">`
Next	(Hidden field) Used to set the page to which the user will be taken when the email has been sent. `<input type="hidden" value="<pbs:prog>/misc?URL=/misc/password_emailsent.pbs" name="Next">`

Buttons

The following code enables **Send** and **Close** buttons. You should only use a close button if you have used a pop-up window for the page.

```
<input type="submit" name="Submit" value="Submit">
<input type="button" name="Cancel" value="Cancel"
onClick="javascript:window.close();">
```

PwdSent.pbo

An email is sent to the user when she requests her password. This email is sent to the email address that is provided.

The **PwdSent.pbo** must be placed in the objects folder.

<%clienthost%>	The IP-address from which the request was sent.
<%password%>	The user's password.
<%to%>	The email address.

pbs adregusertargets

<pbs:adregusertargets>

This tag is used to match user login profiles and criteria from the SAXOTECH Online database to the OAMS database for the purpose of returning ads targeted to the registered user's criteria. It is only necessary to use this tag if you decide not to use SAXOTECH Online's built-in user verification engine.

If you do use this tag, you should place it only on the first page being displayed after a user has been successfully logged in.

For more information, please see the OAMS chapter in the SAXOTECH Online User Manual.

pbs extra_income

Edit user information

This function allows the user to change White Pages information.

This tag is an example of how to output custom user information from the system. In this case, the information is user income. These tags will start with Òextra_Ò followed by the field name of the desired information. This tag is used on the Edit Form.

To let the user update her information, you must add &ForceUserReg=1&Edit=1 to the link that takes her to the White Pages page.

The link should look like this:

```
<a href="frontpage?1&ForceUserreg=1&Edit=1">Edit your info</a><br>
```

Templates

The logon form uses the template **gen_userregedit.pbs**.

The edit template is called **gen_userregeditform.pbs**. The input fields are the same as used in the registration form.

Logon form

This line sets up the form.

```
<form METHOD="POST" action="<pbs:reguser>">
```

Fields

Edit	(Hidden field) Sets the next page to be the edit form. `<input type="hidden" name="edit" value="1">`
Email field	Captures the e-mail address.`<input type="Text" name="email" value="<pbs:#ureg_email#>" size="30" maxlength="40">`
Password field	Captures the password. `<input type="Password" name="pwd" value="<pbs:#ureg_pwd#>" size="20" maxlength="20">`

Edit form

After the user has logged on to the system, the form for editing her current information is shown. The form is built with the following fields.

This line sets up the form.

```
<form METHOD="POST" name="information2" action="<pbs:reguser>">
```

Fields

These fields must be present for the form to work:

```
<input type="hidden" name="Region" value="0000">
<input type="hidden" name="edit" value="1">
<input type="hidden" name="id" value="<pbs:#ureg_id#>">
```

All input fields available in the user registration form can be used in this form as well.

pbs reguserinfo

<pbs:reguserinfo>

When a user is logged in, you can show the user's information.

The tag must be run uncached. Otherwise, the information for the user that generated the cached file will be shown.

Templates

This function can be used on all templates.

Options

<%extra_income%>	To retrieve custom user information use `"extra_"` followed by the custom field names (for example *income*)

Userreginfo.pbo

⚠ This file can support anything contained in the [USERINFO_XXX] section of the session file as a variable

<%User_email%>	The user's email address.
<%User_ID%>	The user ID.
<%UserLevel%>	The User level for forums.
<%User_Name%>	The user name. If the user has not registered with a name, the field will contain `"None None"`
<%UserVerified%>	This field contains the value 1 if the user is a verified forum user.

In the [USERINFO_XXX] section the following user info is available from version 6.5.0sr42

<%USER_fname%>	Shows the first name of the user.
<%USER_lname%>	Shows the last name of the user.
<%USER_mname%>	Shows the middle name of the user.
<%USER_FacebookConnected%>	Shows if the user is connected to a Facebook account.
<%USER_Phone%>	Shows the phone number of the user.
<%USER_address%>	Shows the address of the user.
<%USER_address1%>	Shows address 1 of the user.
<%USER_zip%>	Shows the zip code of the user.
<%USER_city%>	Shows the city that the user has registered.
<%USER_country%>	Shows the country that the user has registered.
<%USER_state%>	Shows the state that the user has registered.

Staff Listing

Staff listing is used to build employee white pages. This function is usually used to display contact information.

- pbs departmentlist
  - DepList.pbo
  - DepListItem.pbo
- pbs departmentoverview
  - DepOverview.pbo
  - DepOverviewGroupHeader.pbo
  - DepOverviewHeader.pbo
  - DepOverviewItem.pbo
- pbs staffinfo
  - StaffInfo.pbo

pbs departmentlist

<pbs:departmentlist>

This function lists all departments registered in the system. This tag can be placed on all templates.

Options

Location	Used to filter the list based on the information in the **Office** field. For more information please see DepOverviewItem.pbo.By default the tag lists all locations. Enter `location="office"` to filter the list so that only records where the office matches will be shown. When you click on this link the resulting list will be filtered as well.
Objectclass	Sets the object class.

DepList.pbo

This file is optional.

<%DepListItems%>	This field lists the items.

DepListItem.pbo

This file is output for each department.

<%Department%>	This field contains the name of the department.
<%Link%>	This field contains the link to the staff list.

pbs departmentoverview

<pbs:departmentoverview>

The tag for displaying the staff list can be placed on any template. The usual way to do it is to place the tag on an overview page like **contact_ovr .pbs.**

> All options are case-sensitive before version 6.9.0sr1 of Saxotech Online.

Options

City	This option is used to filter based on the city field. By default, all cities are listed. Enter `city="xxx"` to limit the list to those staff members with **xxx** in the city field.
Department	This option is used to filter based on the department field. It is now possible to specify a comma-separated list of departments as the value instead of just a single department.
Location	This option is used to filter based on the location field. By default all locations are listed. This value is entered into the office field in the user interface. Enter `location="office"` to filter the list to a single office. When this option is used on your pbs tag you can use the sort field in your edit interface.
Objectclass	Sets the object class.
Namesort	This option makes it possible to sort by Last and then first name. Sortby will overrule this setting.
Sortby	This option makes it possible to sort the staff list by first name, work phone, occupation and more. This is done in the following way: `<pbs:departmentoverview sortby="firstname">` Without the **sortby** parameter specified, the list sorts by last name by default. The following options can be used (case insensitive): • WORKTLF - work phone • STILLING or OCCUPATION • PRIVTLF - private phone • MOBILTLF - cell phone • POSTNR or ZIP • POSTSTED or CITY
Status	Specifies whether the listing shows only active staff (default), or also/instead shows the inactive staff. This is done in the following way: `<pbs:departmentoverview status="active,inactive">` Valid values are: • all • active • inactive or any combination of these. (Any combination of two or three of the values are equivalent to just using "all")

DepOverview.pbo

<%DepOverview%>	This field lists the items. This variable will use the object file called De DepOverviewItem.pbo andpOverviewHeader.pbo (if your staff items have a selected department. Otherwise it is skipped.)

DepOverviewGroupHeader.pbo

Output for each group in the department. This object file is called after DepOverviewHeader.pbo if there are groups under the department. Otherwise it is skipped.

<%DepGroupName%>	The name of the group.
<%DepOverviewItems%>	Outputs all employees in the group.

DepOverviewHeader.pbo

Outputs a header for each department.

<%DepEmail%>	The group e-mail address for the department.
<%DepFax%>	Outputs the department fax number.
<%DepName%>	The name of the department.
<%DepOverviewGroups%>	Outputs the department groups containing all employees and un-grouped employees. DepOverviewGroupHeader.pboIs called if there are groups under the department. Otherwise it is skipped.
<%DepPhone%>	Outputs the department phone number.

DepOverviewItem.pbo

Output for each employee.

<%Address1%>	This field displays address line 1.
<%Address2%>	This field displays address line 2.
<%Address3%>	This field displays address line 3.
<%Address4%>	This field displays address line 4.
<%City%>	This field contains the city name.
<%DepEmail%>	The department email.
<%DepFax%>	The department fax number.
<%DepGroupName%>	The name of the group.
<%DepName%>	The name of the department.
<%DepPhone%>	The department phone number.
<%Email%>	The employee's email address.
<%EmailHome%>	This field contains the personal email address for the employee.
<%Firstname%>	The employee's first name.

<%ImgLink%>	The source address for the image associated with the employee. The maximum size for staff images is 500 x 500 pixels, which is the default. To limit the image to a smaller size, determine the maximum width as in the example below, which limits the width to 100 pixels: `<img src="<%ImgLink%>&MaxW=100" border="0">`
<%Info%>	This field contains the information field.
<%JobTitle%>	The job title.
<%Name%>	The employee's name in the format: Last name, first name.
<%Office%>	This field is used for office location.
<%PhOffice%>	Office phone number.
<%PhDirect%>	Direct phone number (extension number)
<%PhHome%>	Home phone number.
<%PhCellular%>	Cell phone number.
<%PhBeeper%>	Pager number.
<%Phfax%>	This field contains the fax number.
<%Residence%>	This field is used for home location. Typically community/county name.
<%Staffid%>	The ID for the employee. This field is used to provide a link to the details page for the person. It can be used like this: `<a href="/apps/pbcs.dll/personalia?ID=<%StaffID%>"><%Name%></a>`
<%Surname%>	The employee's last name.
<%WebHome%>	This field contains a link to a personal webpage.
<%WebOffice%>	This field contains a link to a work-related webpage.
<%Zip%>	This field contains the Zip/Postal code.

pbs staffinfo

<pbs:staffinfo>

This tag is used to show the details for one person.

Options

Objectclass	Sets the object class.
ID **From version 6.9.0sr18**	Force the tag to show a specific staff member by id

StaffInfo.pbo

This file uses the same fields as DepOverviewItem.pbo, plus the following fields:

<%Firstname%>	The employee's first name.
<%Surname%>	The employee's last name.
<%Field_xxx%> **From version 6.9.0sr2**	Extracts extra fields for the staff member. Replace **xxx**with the name of the field. E.g., `<%Field_Facebook%>`

Tip Functions

The tip functions allow web guests to send links to other internet users. This can include links to stories, classified ads or guide entries in the system.

- Setup tip functions
- External tip page
- pbs article-tiplink
  - ArtTips.pbo
  - ArtTipsText.pbo
  - ArtTipsTextHtml.pbo
  - ClassTipsText.pbo
  - ClassTipsTextHTML.pbo
  - GuideTipsText.pbo
  - GuideTipsTextHTML.pbo
  - Tips.pbo
  - TipsFailed.pbo
  - TipsText.pbo

Setup tip functions

The **TipsText.pbo** file is used when no function-specific file exists. This file has the same fields as ArtTipsText.pbo. If this file does not exist, the system will send an email based on system defaults. It is likely that the default message is in Norwegian.

In order to use tip functions, the following settings must be added to the GENERAL section of the Publicus.ini:

MailID
MailSender
MailServer

if the submit from contains:

<input type="hidden" name="SettingType" value="Prefix">

or SettingType is added to the URL:

apps/pbcs.dll/tips?art=1&AID=/20100810/ARTICLE/8101062/2416/NEWS&SettingType=Prefix

The system will first look for the following in MailSettings section of the publicus.ini

PrefixMailID
PrefixMailSender

External tip page

This tip page is used by the Story, Classifieds and Guide components. The template is called **gen_tip.pbs**.

This table describes what the lines in the form do. Unless stated, all lines are mandatory.

`<form action="<pbs:#action#>" method="POST" name="Tipform">`	Starts the form. The action must be `<pbs:#action#>`.
`<input type="hidden" name="SettingType" value="Prefix">`	Define multiple settings per form. Setup+tip+functions and MailSettings+Section
`<pbs:#hidden#>`	Outputs the hidden system fields.

`<input type="hidden" value="/section/tip_failedsent" name="FailedNext">`	Sets the page to which the web guest will be directed to if the post failed.
`<input type="hidden" value="/section/tip_emailsent" name="Next">`	Sets the page to which the web guest will be directed after the email has been sent.
`<script>document.write(document.Tipform.tittel.value);</script>`	Writes the category name for the section being viewed. Optional.
`<input type="text" name="from" size=20>`	Captures the user's name. Optional, but should be present.
`<input type="text" name="fromemail" size=20>`	Captures the user's email address. Optional, but should be present.
`<input type="text" name="to" size=20>`	Captures the recipient's email address.
`<textarea name="message"cols=20 rows=3></textarea>`	Outputs the message area that is used for a message to the recipient. Optional, but should be present.

`<input type="submit" name="Submit" value="Send">`	Outputs the **Submit** button.
`<input type="hidden" name="HTML" value="1">`	Use HTML. Default is 0. Will look for object file TipsTextHTML.pbo
`<input type="button" name="Cancel" value="Cancel" onClick="javascript:window.close();">`	Outputs a **Close** button. Optional. This feature should only be used if you open the tip form in a separate window.
`</form>`	Closes the form.

For an article tip, please see ArtTipsText.pbo. For a Classifieds tip, please see ClassTipsText.pbo.

pbs article-tiplink

<pbs:article-tiplink>

This tag creates a link to the *Tip a friend* page. There are two versions of the article tip. The form integrated into the page and the pop-up page version. This section covers the integrated tip form. For information about the pop-up version, see External tip page.

The field <%tips%> in ArtTips.pbo triggers the form being displayed. The tip page is described in the chapter External tip page. This tag can be used on all article templates. This tag has no options. The function requires that the **Tips active** checkbox has been marked on the properties page for the category.

Example code
This piece of code will open a new window when a web guest clicks the text *Email this to a friend*.

```
<SCRIPT LANGUAGE="JavaScript1.2">
function NewWindow(height,width,url)
{window.open(url,"ShowProdWindow","menubars=0,scrollbars=1,resizable=1,height="+height
+",width="+width);
}
</SCRIPT>
<a href="javascript:NewWindow(450,300,'<pbs:article-tiplink>');">Email this to a
friend</a>
```

Check field values 1

To check that the **To** field has been filled in, use the following code:

```
onSubmit="javascript:if (this.To.value=='') {alert('You must fill in the recipient
address!'); return false;} "
```

Check field values 2

This function is used to check multiple fields and it is included in the example:

```
<script language="JavaScript">
function checkTipsForm(f) {
 if ((f.To.value=='') || (f.Fromemail.value=='') || (f.From.value=='') ||
(f.Message.value=='')) {
alert('All fields must be filled in!');
return false;
 }
}
</script>
<form action="<%Action%>" method="POST" onSubmit="javascript:return
checkTipsForm(this);">
```

ArtTips.pbo

<%Action%>	Outputs the correct URL for the action of the form. it must be placed inside the form tag like this: `<form action="<%Action%>" method="POST">`
<%from%>	The sender's name.
<%Fromemail%>	The sender's email address.
<%Html%>	If this field has the value 1, the file ArtTipsTextHtml.pbo will be used instead of the normal file. This field should be used as a hidden field.
<%ImgLink%>	The link to the story's picture.
<%Message%>	This text area is used for the message entered by the web guest.
<%Next%>	This hidden field contains the information needed to redirect the web guest to a page after the tip has been sent. To redirect back to the story, use the following code: `<input type="hidden" value="<pbs:link artikkel inherit href>" name="Next">`
<pbs:#hidden#>	Outputs the hidden fields needed by the system.
<%Title%>	This field contains the title of the story and must be placed in a hidden field named "Tittel" like this: `<input name="Tittel" type="hidden" value="<%Title%>">`
<%To%>	The recipient's email address.

ArtTipsText.pbo

This file is the email that is sent to the tip recipient.

<%Category%>	Output category.
<%ClientHost%>	Output Client host
<%From%>	The name entered by the sender.
<%Fromemail%>	The email address provided by the sender.
<%Message%>	The message from the sender.
<%Longlink%>	The full link to the story.
<%Link%>	The short link to the story.
<%Title%>	The story's title.
<%To%>	The recipient's email address.

ArtTipsTextHtml.pbo

This file has the same fields as the normal file. It is used when the flag HTML is set to 1 in the object file.

ClassTipsText.pbo

The tip function allows a web guest to send an email to someone that might be interested in the ad he is viewing. This function is also used to send tips about stories.
The field

```
*<%TipLink%>*
```

in ClassifiedsItem.pbo contains the link to the tip page. The external tip page is used to send the tip.

This file is used for the email sent by the classifieds tip. This file has the same fields as ArtTipsText.pbo.

ClassTipsTextHTML.pbo

The tip function allows a web guest to send an email to someone that might be interested in the ad he is viewing. This function is also used to send tips about stories.
The field

```
<%TipLink%>
```

in ClassifiedsItem.pbo contains the link to the tip page. The external tip page is used to send the tip.

This file has the same fields as ClassTipsText.pbo. It is used when the flag HTML is set to 1 in the object file.

GuideTipsText.pbo

The tip function is also available for guide entries. The field

```
*<%TipLink%>*
```

contains the link to the external tip page.

The email sent to the web guest uses GuideTipsText.pbo. It has the same fields as ArtTipsText.pbo.

GuideTipsTextHTML.pbo

The tip function is also available for guide entries. The field

```
*<%TipLink%>*
```

contains the link to the external tip page.

This file has the same fields as the GuideTipsText.pbo. It is used when the flag HTML is set to 1 in the object file.

Tips.pbo

<%Action%>	Outputs the correct URL for the action of the form. it must be placed inside the form tag like this: `<form action="<%Action%>" method="POST">`
<%from%>	The sender's name.
<%Fromemail%>	The sender's email address.
<%Html%>	If this field has the value 1, the file ArtTipsTextHtml.pbo will be used instead of the normal file. This field should be used as a hidden field.
<%ImgLink%>	The link to the story's picture.
<%Message%>	This text area is used for the message entered by the web guest.
<%Next%>	This hidden field contains the information needed to redirect the web guest to a page after the tip has been sent. To redirect back to the story, use the following code: `<input type="hidden" value="<pbs:link artikkel inherit href>" name="Next">`
<pbs:#hidden#>	Outputs the hidden fields needed by the system.
<%\Title%>	This field contains the title of the story and must be placed in a hidden field named "Tittel" like this: `<input name="Tittel" type="hidden" value="<%\Title%>">`
<%To%>	The recipient's email address.

TipsFailed.pbo

From version 7.2.0sr36, 7.3.0sr1 TipsFailed.pbo will be available when tips fails:

<%MaxNumberRecipientsReached%>	This variable will have the value 1 if the number of recipients are too high
<%LinksNotAllowed%>	This variable will have the value 1 if the message text contains links that are not allowed
<%CaptchaCheckFailed%>	This variable will have the value 1 if captcha validation failed
<%ErrorMessage%>	This contains a short error message based on the failure

TipsText.pbo

The TipsText.pbo file is used when no function-specific file exists. This file has the same fields as ArtTipsText.pbo.

Newsletter Composer Tags

The Newsletter Composer function allows you to send newsletters for which you select the stories. This newsletter can be targeted based on demographic information and on subscription categories.

Setup

The server must be set up to send out newsletters for your website. If this is not done, you can create the newsletters, but they will not be sent. Your server administrator must set up the routine that actually sends the e-mails out to the subscribers.

Newsletter.ini

The file that controls the newsletter is called **newsletter.ini**. This file must be placed in the objects folder. For more information, see the section Newsletter.iniFile.

Templates

These files are used when sending the e-mail to the subscriber. The HTML versions of the templates have HTML added before .pbo, so NewsLetterArticle.pbo becomes NewsLetterArticleHTML.pbo. All templates must be put in the objects folder.

It is important that you keep your lines under 60 characters in order for them to be handled correctly by the email servers.

- Web based management
- Main template
- NewsletterAd.pbo
- NewsletterArticle.pbo
- Newsletter email
- Newsletter ini

Web based management

Web-based management of newsletters lets subscribers manage their subscriptions through a webpage. This can be done on your standard newsletter subscription page. You should name your category something that will make it easy to see that is a newsletter composer category. The value you specify will be shown in the newsletter composer interface

Example code

This code goes on your subscription page.

```
<input type="checkbox" name="Category" value="COMPFOOTBALL" <pbs:#CB_COMPFOOTBALL#>>
Football roundup<br>
<input type="checkbox" name="Category" value="COMPARTS" <pbs:#CB_COMPARTS#>> Arts
newsletter<br>
```

User registration

The web guest must be a registered user of your site. This is because the newsletter uses the same database as the White Pages for user registration. User registration is described in the section User Registration.

Main template

The main template is coded directly in the newsletter composer. The following fields are available.

<%LongDate%>	Outputs the date field.
<%LongDateISO%>	Outputs the date field in ISO format.
<%NewsletterContent%>	Outputs all ads and stories.

<%UnsubscribeURL%>	Contains an URL used to unsubscribe. The unsubscribe link is only valid for newsletters sent based on categories (Newsletter subscribers). It should be put in a conditional like this: `[%{SubscriptionType=0}` `<a href="<%UnsubscribeURL%>">` `Unsubscribe` `</a> ` `%]`

NewsletterAd.pbo

This file is output for every ad.

<%AdText%>	The ad text.
<%AdURL%>	This field contains the value of the URL field on the ad.
<%\Title%>	The ad heading.

NewsletterArticle.pbo

Output for each story. It must be placed in the objects folder.

<%Category%>	The name of the category.
<%\Date%>	The publishing date for the story.
<%ImgLink%>	The link to the main photo attached to the story.
<%Leadin%>	The leadin for the story. To avoid problems with the e-mail server, you should add an extra space after each period. This example also shows how you can crop your leadin to a maximum of 300 characters. `<%Leadin$w(1,300)$r(".=. ")%>`
<%Link%>	The link to the story.
<%Lopenr%>	The story's ID number.
<%\Title%>	The title of the story.

Newsletter email

Getcontent

Retrieves content from another website. You can have more than one `getcontent` tag. The tag can be used with newsletter composer in both the plain text and HTML version of the newsletter. The tag is defined as follows:

```
<newsl:getcontent url="http://www.helloworld.com/apps/pbcs.dll/section?category=newsl>
```

or

```
<newsl:getcontent removehtmlheader=1
url="http://www.helloworld.com/apps/pbcs.dll/section?category=newsl>
```

removehtmlheader	Determines whether HTML header information should be removed. It is also possible to activate the remove for all `getcontent` tags by calling **newsletter.exe** with the parameter `-removehtmlhead`
URL	The URL from which content will be retrieved. If the URL is not valid, the tag will be empty and the newsletter will not be sent.

To activate `getcontent` in the newsletter, the **newsletter.exe** must be called with the parameter

```
-allowgetcontent
```

If the tag fails, the newsletter will not be sent. The application will try to resend the newsletter during the next run.

Newsletter ini

Newsletter.ini file

The file that controls the newsletter (including emails sent using Newsletter Composer) is called newsletter.ini. This file must be placed in the objects
folder. The file has the following sections:

[SETTINGS]

This is the header for the settings section

CharSet	Sets the character set of the email. It is set like this: CharSet=iso-8859-1.
From	Sets the sender of the e-mail. The option is used like this: From=nevedaappeal.com
Language	Sets the correct language for use in the date, for example. The option is used like this: Language=DEN (Danish) Language=SWE (Swedish) Language=GER (German)
ReplyTo	Sets the reply to address for the e-mail. The option is used like this: ReplyTo=sendereditor@tahoe.com
Sender	Sets which email address the newsletter is sent from. The option is used like this: Sender=info@domain.com. ⚠ This must be a valid mail addresss. Not only a name

[System]

This is the header for system section. It must be present.

Antall	Sets the maximum number of stories sent from a single category
DateFormat	Sets the date format. This field uses the standard SAXOTECH Online date formatting options. The option is used like this: DateFormat=dddd, mmmm dd, yyyy

Sniffcategories	By default the Sniff Search looks in all categories for the system. To limit Sniff Search categories, use this option like this: SniffCategories=NEWS,SPORT,...
Subject	Sets the subject of the e-mail. The option is used like this: Subject=Your email newsletter from nevadaappeal.com
UseExpandedLink	If enabled(UseExpandedLink=1), links are displayed as: /apps/pbcs.dll/article?AID=/20100101/NEWS/12345678
UrlRewrte	If enabled(URLRewrite=1), links are displayed as: /article/20100101/NEWS/12345678 This setting works only with UseExpandedLink
RelProfile	If enabled(RelProfile=1), the most relevant profile is added to the url and templates. This setting works only with UseExpandedLink
ExpNodes	If enabled(ExpNodes=1), ExpNodes is added to the url and templates. ExpNodes is used to navigate trough Profile tree menu. This setting works only with RelProfile and UseExpandedLink
URL	The URL is used to set the base URL for your site and is required for the <%link%> variable to return the correct domain. The option is used like this: URL=http://www.nevadaappeal.com

Omniture

SAXOTECH Online web site statistics information is provided through a web analytics engine called SiteCatalyst powered by Omniture.

- Setup Omniture
  - Procedure to Implement Omniture
    - Test Environment
    - Implementing Omniture
- pbs omniturecode
  - Omniture.pbo
  - URL action

Setup Omniture

To use this tag, the following settings must be made in the Publicus.ini file in the OMNITURE section:

```
[OMNITURE]
UseOmnitureTag=1
SuiteID=saxotechcom
BuildHierarchy=0
CookieDomainPeriods=2
```

The `UseOmnitureTag=1` setting disables the automatic insertion of the Omniture JavaScript. Instead, it inserts Omniture JavaScript wherever an Omniture tag is found in the templates.

Please note that the following actions (on other tags) are not affected by this setting: Tip a Friend, AdClick, PollClick and NewsLetterSignup. They will generate statistical information anyway.

Because Omniture SiteCatalyst will only generate statistics if there is Omniture JavaScript on the pages, we recommend not using this feature except in special circumstances.

Procedure to Implement Omniture

This documentation demonstrates how to implement Omniture on a live site. On multiple occasions it will reference the Omniture.pbo and other file documentation under Confluence.

What is Omniture?

(Taken from here) Omniture is the leader in web analytics. The Omniture® Online Marketing Suite is the leading solution to optimize ad spend and conversion. It includes a comprehensive portfolio of applications for visitor acquisition, conversion, online analytics, and multichannel analytics built on an open business analytics platform.
Adobe systems acquired Omniture on October 23, 2009.

Where can I access the actual data?

You can login to Omniture and view the site statistics by going to the following link:

https://my.omniture.com/login

The login and password can be provided upon request by the Senior Developer or other senior staff members.

Test Environment

Implementing Omniture

Test Environment

In order to properly implement Omniture across the site, it is recommended that you setup a test environment so you can better troubleshoot any issues that arise.

Enable Omniture object files

To have your Omniture code display, it must be enabled in the Publicus.ini. Refer to the Setup Omniture page for more information.

Install FireBug

You will need to install Firebug in Firefox in order to install Omnibug.

Install OmniBug

You will need to install Omnibug to properly test Omniture

Creating a test template

Create a new file name under your site's template directory. Name it *gen_omnituretestTEMPLATE*. An example is below:

```
\\s260608at1vw701\repository\Web\GE\mal\gen_omnituretestTEMPLATE.pbs
```

Use the following code inside this template:

```
<h1>\mal\gen_omnituretestTEMPLATE.pbs</h1>
<h3>test00</h3>
<pbs:!macro name="OmnitureCall" />
```

As you can see a Macro file may also need to be created. This macro will contain any test variable that you may wish to add to your Omniture code in addition to the tag that will call the script:

```
<pbs:omniturecode ObjectClass="2">
```

As you can see this tag will call *Omniture2.pbo* under the _\templates\objects_ folder. If you have an Omniture object created already, go ahead and place it under that directory. Otherwise use the sample code from the omniture.pbo page. Most of the code is JavaScript doing calls to the Omniture API.

Testing Omniture

You will need to pick a specific server to do your testing. This way your changes do not reflect across the whole site. You will need to know the cluster where the site is located so you can pick a particular server to test on. For example, if we are testing www.GoErie.com, their Site code is GE. This falls under the SX cluster. From here we can go to the appropriate text file containing the server values.

Getting the server variables

To get the server variables (or names), go to the following directory:

```
\\s260608at1vw776\e$\Files
```

Using your site code as a reference point, look for the following file:

```
[CLUSTERCODE]web.txt
```

For example if you want to get the servers for www.GoErie.com (SITECODE=GE, CLUSTERCODE=SX), then you will look for sxweb.txt. Opening sxweb.txt will give you the following servers:

S260608AT1VW721, S260608AT1VW722, S260608AT1VW723, S260608AT1VW724, S260608AT1VW725, S260608AT1VW726, S260608AT1VW727, S260608AT1VW728, S260608AT1VW729, S260608AT1VW730, S260608AT1VW731, S260608AT1VW732, SXWEB13, SXWEB14, SX1oWEB015, SX1oWEB016, SX1oWEB017, SX1oWEB018

Pick any server that starts with "S260608AT1VW". In this case lets pick S260608AT1VW727.

Access the specific server in your browser

Now that we have a specific server selected, we need to find its web alias to access it directly in your browser. To do this, enter the following in your elevated command prompt while logged in to 776:

```
STAR S260608AT1VW727 SITE GE GETVAR SERVERBINDINGS
```

This will give you the following information:

```
S260608AT1VW727/GE: GEweb.sx.atl.publicus.com - www.goerie.com - goerie.com -
wap.goerie.com - GEweb07 - www.gocrawfordcounty.com - gocrawfordcounty.com
```

What we are looking for is *GEweb07*. You can now access the website using that specific server with:

```
http://geweb07/
```

Remember that you can only access a specific server from within 776. To access the actual template you created earlier type the following:

```
http://geweb07/apps/pbcs.dll/section?template=omnituretestTEMPLATE&nocache=1
```

Recycle to view changes

Whenever you are making changes to the code you are testing, you need to recycle the app pool for that specific server. To do this, open an elevated command prompt and enter the following:

```
star [SPECIFIC SERVER] pool [SITECODE] recycle
```

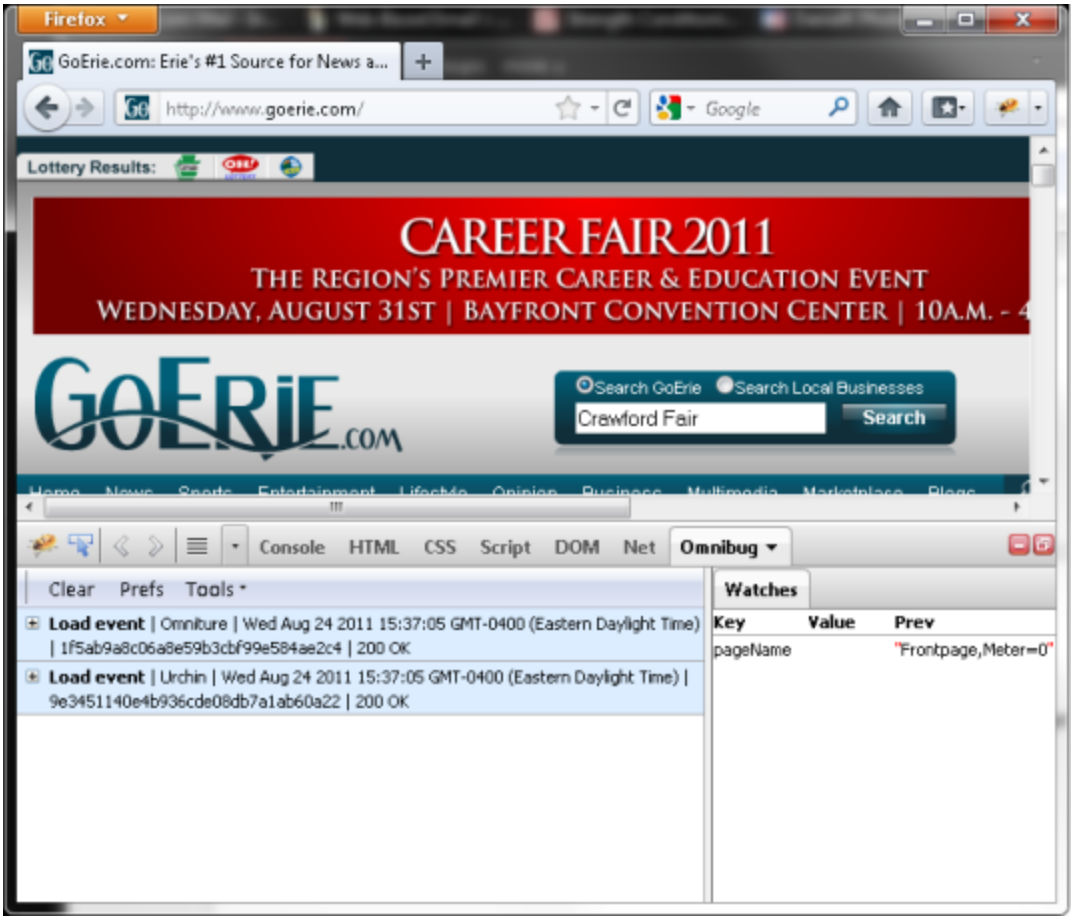
For example:

```
star S260608AT1VW727 pool ge recycle
```

Keep in mind that you can recycle the app pool so many times before it shuts down. If it shuts down, run a *start* command instead of a *recycle*.

Using Omnibug

When you implement Omniture on a template file, you will get the following view:



You can expand the view to verify the contents.

Implementing Omniture

Once you are satisfied with the results of your object file, you will want to move the macro call to an include file that is being called on all *articlexx.pbo* files:

```
<pbs:macro name=OmnitureCall ArtNo="<%@StoryID%>" after>
```

The *after* parameter sets the macro to be called at the end of the pbs parsing procedure.

To ensure that Omniture runs on article files only, you want to pass in the *<%StoryID%>* parameter through the macro for conditional testing. Modify the macro you are using to call the Omniture code as in the following:

```
[%{ArtNo}<pbs:omniturecode>%]
```

Adding extra fields

If you need to use article variables inside the Omniture call, you will have to create a global variable inside all *articlexx.pbo* files and pass the variable through the macro as mentioned above. To do this follow the procedures to create Global variables. Make sure to make a backup of the files you are working on prior to modifying any article or include files.

Recycle the site

When the desired changes have been made, you will need to recycle the site using the star command. To do this, do the following on an elevated command prompt:

```
star [CLUSTERCODE]web.txt pool [SITECODE] recycle
```

For example:

```
star sxweb.txt pool ge recycle
```

Recache the site

After the recycle you will want to remove any old cached files so the new changes take effect. To do this you will need to go to the status page of the website. For example, *www.goerie.com\status*. Check if the *Cache Type* is *File*. You will also see a parameter called *Cache Path*. Copy this path and use it for the following command under an elevated command prompt:

```
DEL \\[CACHEPATH]\*.* /q
```

This may take between 5 to 30 minutes depending on the amount of files cached.

pbs omniturecode

<pbs:omniturecode>

In order for Omniture to gather website statistics, it must find Omniture JavaScript on the pages it should analyse. If you would like to determine on which pages and where on the pages to insert Omniture JavaScript, you should use the Omniture tag pbs omniturecode.

Checks will be made to ensure that the code will only be inserted once per page at the bottom of the page.

ObjectClass	Specifies what object file design class to use.

Example:

```
<pbs:omniturecode ObjectClass="1">
```

Omniture.pbo

The following list of parameters is available for use in the object file.

<%*s_codeFileLocation%>*	Contains the path to the s_code.js file provided by Omniture
<%*Account%>*	The Omniture Account name, taken from the Publicus.ini value.
<%CookieDomainPeriods%>	Number of Periods (.) in the domain. This is usually 2 and is set in the publicus.ini file.
<%PageName%>	Contains the pagename, generated by the DLL.
<%Server%>	Contains the server name that generated the content.
<%Channel%>	This has the current category.
<%PropX%>	Props are used to store extra information about the current page. Omniture supports up to 50 props, however, they do slow down the Sitecatalyst site a bit, so pr default they are not all enabled. It is important not to use the same props for different values. We suggest that the customers uses prop 10 to 20. Saxotech currently use the first 5. Prop 1 : Pagetype Prop 2 : Printed / Emailed Prop 3 : Profile Prop 4 : Author Prop 5 : Tracker Prop 6 : Errormessage Prop 7 : Search String, this is only used on the search action. **(from version 6.8.0sr24)** Of cause these should be set on custom pages, where the dll does not set this automatically.90

<%Events%>	Events are used with Products. The events variable is used to describe the state of the products. Saxotech currently use: event1 : Legacy banner system, Impression event2 : Legacy banner system, Clicks event3 : Poll mini Impression event4 : Poll mini vote event5 : Newsletter subscribe This is used, because the best way of tracking information across different pages is to tag them as Products in an event state.
<%Products%>	Omniture have extensive tracking for web shops. The best way of tracking elements across different pages is to add them to the products variable. We use this for the legacy ads, pollmini and newsletter subscribes
<%eVarX%>	EVars is currently not used by Saxotech. They can be used to track users paths around a web shop.
<%HierX%>	Hierarchy variables are used to show a drill down report of the site. We use them for the category Tree. This means it is possible to see the amount of pageviews on the main category and then drill down through subcategories and the stories in these categories.
<%PageType%>	The type of content on the current page (e.g. Story, section, Frontpage).
<%isAd%>	The legacy ad system handles the omniture request slightly different. It is not advisable to change this part of the templates, although it is possible.

Default Template used by Saxotech, it should be noted, that this may change from version to version: Download

```
[%{!isAd}<script language="JavaScript" type="text/javascript"
src="<%s_codeFileLocation%>"></script>
<script language="JavaScript" type="text/javascript">%][%
s.Account="<%Account%>"%][%
s.cookieDomainPeriods="<%cookieDomainPeriods%>"%][%
s.pageName="<%PageName%>"%][%
s.server="<%Server%>"%][%
s.channel="<%Channel%>"%]
/* Traffic Variables */[%
s.prop1="<%Prop1%>"%][%
s.prop2="<%Prop2%>"%][%
s.prop3="<%Prop3%>"%][%
s.prop4="<%Prop4%>"%][%
s.prop5="<%Prop5%>"%]
/* E-commerce Variables */[%
s.events="<%Events%>"%][%
s.Products="<%Products%>"%][%
s.eVar1="<%eVar1%>"%][%
s.eVar2="<%eVar2%>"%][%
s.eVar3="<%eVar3%>"%][%
s.eVar4="<%eVar4%>"%][%
s.eVar5="<%eVar5%>"%]
/* Hierarchy Variables */[%
s.hier1="<%Hier1%>"%][%
s.pageType="<%PageType%>"%]
[%{!isAd}/************* DO NOT ALTER ANYTHING BELOW THIS LINE ! **************/
var s_code=s.t();
if(s_code)document.write(s_code)
</script>
<noscript>
<img src="http://grndsinchant.112.2O7.net/b/ss/grndsinchant/1/H.2--NS/0"height="1"
width="1" border="0" alt="" /></noscript>%]
```

Omniture have further information about this, however, login to Omniture Sitecatalyst may be needed to access these documents:

Javascript Debugger (Used to check the variables set on the page): Whitepaper

Sitecatalyst Implementations Guide: http://microsite.omniture.com/t2/pdf/en/sitecatalyst_implementation_guide.pdf

URL action

In order to exclude Omniture JavaScript from the response of certain actions, you can add `&omniture=0` to the URL. This will work for all responses of actions into which the Omniture JavaScript is normally inserted.

In order to exclude Omniture JavaScript from the response of the story and section actions, you can add `&mime=xml` to the URL. For example, this can be used if NITF Stories or RSS feeds are requested.

US Weather

To be able to show weather on your website, you need to have a feed from a weather provider. SAXOTECH Online only offers the tools to display the weather information on your site. We do not provide weather information.

The system does not know which measurement system is being used. The system does not see the difference between Fahrenheit and Celsius, so you must use the same measurement system as your weather provider.

There are multiple sets of weather tags. The weather tags starting with "Accu" are only used in the US.

- pbs accuwforecast
  - AccuWForecast.pbo
- pbs accuwcurrent
  - AccuWCurrent.pbo
- pbs accuwname
- pbs accuwmap
  - AccuWMap.pbo
- pbs accuwoptions

pbs accuwforecast

<pbs:accuwforecast>

The forecast function displays the forecasted weather.

Options

Label	Used to select the geographical location code.
Dateformat	Sets the date format. This option follows the default SAXOTECH Online behavior and can be set like this: `dateformat="h:mmam/pm"`
Class	Sets the object class for the tag.

Part	Determines which record in the forecast table is used. Enter
	`part=0`
	to show the first record. Enter
	`part=1`
	to display the second record. The information displayed depends on your weather provider. Enter
	`part=-1`
	to display all records.

Local	Determines which table will be used in the database. If the weather provider delivers more details for your local region, you can use this table instead by entering `local=1` If this option is not set, the forecast will use the default national table.
World	Used to select data from outside the US. The option is used like this: `world=1`

AccuWForecast.pbo

<%Updated%>	The update time for the weather.
<%CurrLabel%>	The forecast label.
<%Name%>	The forecast name.
<%Part%>	The part ID for the forecast.
<%Icon%>	The icon number for the weather. It is usually used like this: `<img src="/graphics/weather/icons/<%Icon%>.gif">`
<%ShortDesc%>	The short description of the weather. This is usually a single word like sunny, foggy or overcast.
<%DayDesc%>	The description of the day.
<%High%>	The projected high temperature in Fahrenheit.
<%HighCelsius%>	The projected high temperature in Celsius.
<%Low%>	The projected low temperature in Fahrenheit.
<%LowCelsius%>	The projected low temperature in Celsius.
<%WindSpeed%>	The wind speed.
<%WindDir%>	The wind direction.
<%WindGusts%>	The projected wind gusts.
<%Humidity%>	The humidity value.
<%Dewpoint%>	The dew point.
<%Barometer%>	The barometric pressure.
<%Visibility%>	The visibility information.
<%State%>	The state code for the entry being viewed.
<%MapCode%>	The map code for the entry being viewed.

pbs accuwcurrent

<pbs:accuwcurrent>

This function gives you the current weather.

The tag must be used like this:

```
<pbs:\!accuwcurrent label="TOL" class=1>
```

Options

Label	This option is used to select the geographical location code. The code must be in the system's option table, and Accuweather must transfer information for the code to the server. You can also set the label in the URL by adding `&Label=XXX` to the URL.
Locklabel	Used to lock the label for the tag so it cannot be overridden from the URL. This is useful when you have a weather symbol in the header that should not change. By default this option is not used. To use locklabel enter `locklabel=xxx` in the tag instead of Label.
Class	Sets the object class for the object file.
World	Used to select data from outside the US. The option is used like this: `World=1`
StateInfo	Makes the tag list all cities in the selected state using the Label option. By default this option is not used. To use StateInfo enter `Stateinfo=1`
Selection	Used to specify multiple city codes. Multiple codes must be specified using a semicolon separated list.

AccuWCurrent.pbo

Not all vendors of weather content provide all the fields listed.

<%Updated%>	The update time for the weather.

<%CurrLabel%>	The current label.
<%Name%>	The current name.
<%ShortDesc%>	A short description of the weather. This is usually a single word like sunny, foggy or overcast.
<%Icon%>	The icon number for the weather. It is usually used like this: `<img src="/graphics/weather/icons/<%Icon%>.gif">`
<%CurrTemp%>	The current temperature.
<%FeelsLike%>	The feels-like temperature.
<%Humidity%>	The humidity value.
<%Dewpoint%>	The dew point.
<%Barometer%>	The barometric pressure.
<%WindSpeed%>	The wind speed.
<%WindDir%>	The wind direction.
<%WindCurr%>	The current wind speed.
<%WindGusts%>	The current wind gusts
<%Visibility%>	The visibility information.
<%Sunrise%>	The sunrise time.
<%Sunset%>	The sunset time.
<%State%>	The state code for the entry being viewed.
<%MapCode%>	The map code for the entry being viewed.

pbs accuwname

<pbs:accuwname>

This function outputs the name associated with a label.

Options

Label	The name of the location for which you would like to show weather. This is the same label you use to show current conditions and forecasts.
Country	Sets the country code for the label. By default the country code is The United States. Other country codes can be used.**Example:** C019 = Bermuda

pbs accuwmap

<pbs:!accuwmap>

This function allows you to display maps provided by the weather service.

The{{<pbs:!accuwmap>}} tag can be placed on all templates.

Options

Maplabel	Selects the map from the database. Which maps are available depends on the weather service you use. Please see the list provided by your weather provider. The following maps are provided by AccuWeather: • LOCAL_RAD - Local radar. • REGRAD - Regional radar. • USRAD - US Radar. • TODHI - Today's high temperatures. • TONLO - Tonight's low temperatures. • REGSAT - Regional satellite. • USSAT - US Satellite.
Dateformat	Sets the date format. This option follows the default SAXOTECH Online behavior and can be set like this: `dateformat="h:mmam/pm"`
Maxwidth	Sets the maximum width of the map in pixels.

AccuWMap.pbo

<%Map%>	Contains the path to the map. It is use in an image tag like this: `<img src="<%Map%>" alt="" border="0">`

pbs accuwoptions

<pbs:accuwoptions>

This tag can be used to build a search form that allows web guests to browse states and cities.

The {{<pbs:accuwoptions> }}tag can be used on all pages. The tag must be used uncached.

Options

States	This option makes the tag list states.
Cities	This option makes the tag list cities.
States	This option makes the tag list states.
States	This option makes the tag list states.
Cities	This option makes the tag list cities.

Example search form

```
<form name="weathersearch" action="<pbs:prog>/section?Category=WEATHER2"
method="POST">
<input type="hidden" name="LastState" value="<pbs:!#State#>">
<input type="hidden" name="MapLabel" value="<pbs:!#MapLabel#>">
<table border=0>
<tr>
<td><b>Choose state: </b></td>
<td colspan=2>
<select name="State" OnChange="javascript:document.weathersearch.submit();">
<pbs:!accuwoptions type=states>
</select>
</font>
</td>
</tr>
<tr>
<td><b>Choose city: </b></font></td>
<td colspan=2>
<select name="Label" OnChange="javascript:document.weathersearch.submit();">
<pbs:!accuwoptions type=cities>
</select>
</td>
</tr>
</table>
</form>
```

European Weather

These tags are used to present weather information provided by Storm Weather Services, Cirrus, DNMI and DDMI.

Templates

The European Weather tags can be used on all templates.

- pbs weatherforecast
  - WeatherForecast.pbo
  - WeatherForecastColItem.pbo
  - WeatherForecastItem.pbo
  - WeatherForecastRowItem.pbo
- pbs weathergraph
  - WeatherGraph.pbo
- pbs weatherlastupdate
- pbs weatherobservations
  - WeatherObservation.pbo
  - WeatherObservationItem.pbo
- pbs weatherpicture
  - WeatherPicture.pbo
- pbs weathertext
- pbs weathertidesunmoon
  - WeatherTideSunMoon.pbo
- Storm weather service

pbs weatherforecast

<pbs:weatherforecast>

This tag can be used on all templates.

The <pbs:weatherforecast> tag can be used in the following manner:

```
<pbs:weatherforecast days=4 location=64 site=TA>
```

Options

Days	Determines for how many days into the future the tag should show forecasts.
Location	Used to select the location ID for the information.
Site	This option is used to set the site code from which the weather data will be retrieved. This option is only to be used if you are retrieving information from another site.

WeatherForecast.pbo

<%WeatherForecastContent%>	Lists all the content.

WeatherForecastColItem.pbo

If no region information is used, the weather forecast uses these column and row tags.

<%WeekDay%>	Shows the day of the week.

WeatherForecastItem.pbo

<%WeekDay%>	The day of the week.
<%WCode%>	
<%MaxTemp%>	The maximum temperature.

WeatherForecastRowItem.pbo

If region information is used, the weather forecast uses this tags.

<%Name%>	The region name.
<%ColContent%>	Shows the content of the column.

pbs weathergraph

<pbs:weathergraph>

⚠ Averages are stored on the year 9999 in the table LocationForecast.

This tag can be used on all templates.

Options

Objectclass	Sets the object class for the tag.
Site	Sets the site code from which the weather data will be retrieved. This option is only to be used if you are retrieving information from another site.
Location	Used to select the location ID for the information.
Dateformat	Sets the date format for the graph. It uses default SAXOTECH Online formats. By default, the format is "d.m."
Dateoffset	Sets the offset from today's date.
Days	Sets how many days the graph should show. The default is 30 days.
Type	Sets the type of graph.
Precip	This option sets the graph to show precipitation.
Temp	With this setting the graph shows temperature. This is the default.

WeatherGraph.pbo

<%ImgLink%>	Contains the link to the graph image.

pbs weatherlastupdate

<pbs:weatherlastupdate>

This tag is used to show the latest time that weather information was updated. This tag can be used on all templates.

Options

Format	Sets the date format. This option uses the standard SAXOTECH Online method of formatting dates.
Region	Used to select the region for which you would like to show the update. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.

Type	By default, this is not used and the tag shows the update time from either WeatherRegionForecast or WeatherLocationForecast, depending on which was updated most recently. Set `type=observations` to force the tag to show either WeatherRegionForecast or WeatherLocationForecast.

pbs weatherobservations

<pbs:weatherobservations>

This tag is used to report weather observations. This tag can be used on all templates.

Options

Region	Used to select the region for which you would like to show observations. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Site	Sets the site code from which the weather data will be retrieved. This option is only to be used if you are retrieving information from another site.

WeatherObservation.pbo

<%RegionName%>	The region name.
<%ObservationItems%>	Lists all the items.

WeatherObservationItem.pbo

<%Name%>	The region name.
<%WCode%>	
<%MaxTemp%>	The maximum temperature.
<%WdDir%>	The wind direction.
<%WdSpeed%>	The wind speed.

pbs weatherpicture

<pbs:weatherpicture>

This tag is used to display a weather image. This image can be a map, satellite image, or other image provided by the weather service. This tag can be used on all templates.

Options

Region	Used to select the region for which you would like to show the picture. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Width	Sets the width of the weather picture.

OnclickUrl	Used to open up a larger version of the image. It is used like this: `<pbs:weatherpicture region=8 width=235 onclickurl="javascript:NewWindow(536, 434,'<%imgurl%>&maxw=500')">`

WeatherPicture.pbo

<%Picture%>	This field contains the picture URL.

Example

```
<%Picture%><br>
```

pbs weathertext

<pbs:weathertext>

This tag is used to retrieve the weather forecast text and the description of the current weather. This tag can be used on all templates.

Options

Region	Used to select the region for which you would like to show the forecast. Which region codes match which areas is dependant on the server configuration. Contact the server administrator for a list of relevant region codes.
Dateoffset	Determines for how many days into the future you would like to show the forecast.
Site	Sets the site code from which the weather data should be retrieved. This option is only to be used if you are retrieving information from another site.

pbs weathertidesunmoon

<pbs:weathertidesunmoon>

This tag can be used on all templates.

WeatherTideSunMoon.pbo

<%TideHigh%>	The time when the tide is highest.
<%TideLow%>	The time when the tide is lowest.
<%Sunrise%>	The time the sun rises.
<%Sunset%>	The time the sun sets.
<%Moonrise%>	The time the moon rises.
<%Moonset%>	The time the moon sets.
<%MoonPhaseType%>	The phase of the moon.
<%MoonPhaseTime%>	The time of the moon phase.

Storm weather service

Storm weather service provides a different type of service and has its own separate tags.

Object files from detskjer.no:

```
<pbs:weatherpicture region=8 width=235
onclickurl="javascript:NewWindow(536,434,'<%imgurl%>&maxw=500')">
<pbs:weathertext region=3 site=TA><br>
<pbs:weatherlastupdate format="dd. mmmm yyyy kl. hh:nn">
<pbs:weatherobservations site=TA region=1>
<pbs:weatherlastupdate format="dd. mmmm yyyy kl. hh:nn" region=8 type=observations>
<pbs:weathertext region=8 dateoffset=3><br>
<pbs:weatherforecast days=4 location=64 site=TA>
```

Horoscopes

The horoscope function is used to display horoscope data imported from a horoscope provider.

- pbs horoscope

pbs horoscope

<pbs:horoscope>

Template

The tag can be put on all templates. The example uses the template **horoscope_ovr.pbs**.

Options

Category	The category option is used to select the data for the different signs: • Today - This option makes the tag show the extended horoscope for today. • Aquarius - This option makes the tag show the horoscope for Aquarius. • Pisces - This option makes the tag show the horoscope for Pisces. • Aries - This option makes the tag show the horoscope for Aries. • Taurus - This option makes the tag show the horoscope for Taurus. • Gemini - This option makes the tag show the horoscope for Gemini. • Cancer - This option makes the tag show the horoscope for Cancer. • Leo - This option makes the tag show the horoscope for Leo. • Virgo - This option makes the tag show the horoscope for Virgo. • Libra - This option makes the tag show the horoscope for Libra. • Scorpio - This option makes the tag show the horoscope for Scorpio. • Sagittarius - This option makes the tag show the horoscope for Sagittarius. • Capricorn - This option makes the tag show the horoscope for Capricorn.

TV Guide

The TV Guide Component is used to display the TV guide. To use this component you need a feed from an information provider.

You should use a style sheet when you build the TV guide. If not, your pages will be very large. The examples in this documentation use style sheets.

The style sheet used by the TV guide is called tv.css and is located in the folder* /includes.*

- pbs tv_getoptions
- pbs tv_LineUpData
  - TVLineUpSelect.pbo
  - TVLineUpZipSelect.pbo
- pbs tv_programinfo
  - TvPrograminfo.pbo
- pbs tv_schedule
  - TVSchedule.pbo
  - TVScheduleHeaderRow.pbo
  - TVScheduleRow.pbo
  - TVScheduleRowCell.pbo

pbs tv_getoptions

<pbs:tv_getoptions>

This tag extracts information from the database.

Template

The search form can be put on any template. The results page must be set to the page used for showing the line-up.

The example is placed on the template **tv_ovr.pbs**.

These hidden fields are used to store information when the web guest browses the TV Guide:

```
<input type="hidden" name="Provider" value="<pbs:!#Provider#>">
<input type="hidden" name="LineUp" value="<pbs:!#LineUp#>">
<input type="hidden" name="Tier" value="<pbs:!#Tier#>">
<input type="hidden" name="Interval" value="<pbs:!#Interval#>">
<input type="hidden" name="StartTime" value="<pbs:!#StartTime#>">
<input type="hidden" name="Genre" value="<pbs:!#Genre#>">
```

The two JavaScripts in the example form are to manage the selections.

Options

Type="Times"	Extracts a list of <option> fields, which is used to select the start time for the lineup. The tag is used like this: `<pbs:tv_getoptions type="Times">`
Type="TVDates"	Extracts a list of <option> fields, which is used to select the date for the lineup. The tag is used like this: `<pbs:tv_getoptions type="TVDates">`

Type="TVChannels"	Extracts a list of <option> fields, which is used to select a single channel. The tag is used like this: `<pbs:tv_getoptions type="TVChannels" Provider="OH60418" LineUp="R" Tier="2" any=1>`

The other options are there to provide defaults. Please see pbs tv_schedule for more information.

pbs tv_LineUpData

<pbs:tv_LineUpData>

Before web guests can view TV listings, they must select their geographical location and TV provider. The TV provider is the cable provider, satellite network, and so on. The template is called **gen_lineup.pbs**. This template must be accessed with **/apps/pbcs.dll/tv_lineup.**

The tag must be used like this:

```
<pbs:!tv_LineUpData useobjects=1>
```

Options

Useobjects	Enables the use of object files.

TVLineUpSelect.pbo

Used to select the provider.

<%Change%>	Stores the previous selection.
<%LineUps%>	Outputs `<option>` fields for the drop-down.

TVLineUpZipSelect.pbo

This form is used to send the zip code to the server. Please see the example code for the form.

<%Change%>	Stores the previous selection.
<%ZipCode%>	Outputs the zip code that was previously entered.

pbs tv_programinfo

<pbs:tv_programinfo>

This page is usually presented as a pop-up window from the TV Guide. Uses the template **X_ovr.pbs**, where X is the category specified in the link to the page. The example uses Tv1_ovr.pbs.

The tag is used in the following manner:

```
<pbs:!tv_programinfo useobjects=1>
```

Options

Useobjects	This option makes the tag use object files.

TvPrograminfo.pbo

<%Actors%>	The list of actors for the program.
<%Crew%>	Contains information about the creators of the program.
<%Description%>	A description of the program.
<%ProgramType%>	The name for the type of program. This name matches the fields in the legend in the main listing.
<%Showtimes%>	The show times for the program.
<%\Title%>	The name of the program.
<%TVRating%>	The age rating.

pbs tv_schedule

<pbs:tv_schedule>

After the web guest has selected her zip code and provider, she will be directed to the page that shows the TV line-up. This page uses the **xxx_ovr.pbs** based on the category selected in the form action. The example uses Tv_ovr.pbs.

The tag is used in the following way:

```
<pbs:!tv_schedule Provider="OH60418" LineUp="R" Tier="2" Interval=120 useobjects=1
width="625">
```

Options

Interval	Determines for how many minutes the TV schedule should be output.
LineupTier	Sets the default lineup.
Provider	Sets the default provider. This option is only used if the page is opened without any parameters.
Useobjects	Enables object files.
Width	Sets the width of the output.

These object files are used both for displaying the line-up for multiple channels and for a single channel line-up.

TVSchedule.pbo

Used to output the entire schedule.

<%ColCount%>	Used by the system to set the width of the columns for the different TV programs.
<%Header%>	The content of the file TVScheduleHeaderRow.pbo
<%Legend%>	The program legend.

<%ScheduleContent%>	Outputs the schedule.
<%StationNum%>	The station number.
<%Width%>	The width set in the tag.

TVScheduleHeaderRow.pbo

Used to output the header for the listing.

<%CellInfo%>	Outputs the cells with the time headers, or the name of the channel if you are viewing the listing of one channel.
<%StationNum%>	Controls the output of either the text "channel" or "time".

TVScheduleRow.pbo

Output for each row in the listing.

<%CellInfo%>	Outputs the information about each program.
<%ChannelCallSign%>	The channel call sign.
<%ChannelNum%>	The channel number.
<%ShowStart%>	The start time of the program when listing all programs for a single channel.
<%StationNum%>	The station number. It is used when linking to the page for listing all programs for a single channel.

TVScheduleRowCell.pbo

<%Category%>	The program's category.
<%CC%>	This field has a value if the program is Closed Captioned.
<%ChannelID%>	The channel ID. This information is used to link to the details page for the program.
<%ProgramID%>	The program ID. This information is used to link to the details page for the program.
<%ProgramType%>	Sets the correct code for the style sheet.
<%Rating%>	The program's rating.
<%SchedID%>	The schedule ID. This information is used to link to the details page for the program.
<%Span%>	Sets the width of the cell.
<%\Title%>	The title of the program.
<%TooShort%>	Contains a value if the cell is too short to show the program information.

Movie Guide

Movie Guide is a module that allows you to show information about cinemas in your region. This is not the same as the Cinema Module, which is based on the events calendar, and in which theaters must enter the information themselves. This movie guide requires that you have an information feed from a content provider.

- Link to own reviews
- pbs mv_movieinfo
    - MVMovieInfo.pbo
    - MVMovieInfoImageItem.pbo
    - MVMovieInfoImageRow.pbo
    - MVMovieInfoReviewItem.pbo
- pbs mv_schedule
    - MVSchedule.pbo
    - MVScheduleChainFooter.pbo
    - MVScheduleChainHeader.pbo
    - MVScheduleMovieItem.pbo
    - MVScheduleTheaterFooter.pbo
    - MVScheduleTheaterHeader.pbo
- Search form for movie guide

Link to own reviews

To attach your own reviews, use the variable

```
#filmid#
```

in the keyword portion of the **Newslist** tag. This will look for stories with the corresponding film ID as a keyword. You can combine the `#filmid#` variable with other keywords if necessary.

```
<pbs:newslist category=XXX keyword="#filmid#" days=90 count=5>
```

pbs mv_movieinfo

<pbs:mv_movieinfo>

This function uses the template **X_ovr.pbs**, where X is specified in the tag linking to the page. The example uses mv1_ovr.pbs.

Options

ImageListColumns	Controls how many columns the images should span. The option is used like this: `ImageListColumns=4`
ImageListStart	Used to skip a number of images. This option is typically used when you use an extra page to display images. The option uses the first picture that the page should show as a parameter like this: `ImageListStart=15`
Objectclass	Sets the object class.

Useobjects	Enables the use of object files.

MVMovieInfo.pbo

This file is the mail file for the **Movieinfo** tag.

<%Actors%>	The name of the actors.
<%Caption%>	The caption for the first image. `<%Caption2%>` contains the caption for the second image, and so on.
<%Description%>	The description of the film.
<%Distributors%>	The name of the distributor.
<%FilmID%>	The unique ID for the film. The ID can be used in links.
<%Genres%>	The genre information.
<%HomePage%>	The URL for the film's homepage.
<%Images%>	Outputs MVmovieinfoimagerow.pbo
<%ImageCount%>	The number of images connected to the movie.
<%ImageLink%>	The URL for the first image. The field `<ImageLink2%>` contains the URL for the second image, and so on.

<%ImageRef%>	The reference code. This is used when you want to use the zoom function on a picture. It is used like this:
	
	`<%pbcs.dll%>/misc?url=/templates/zoom.pbs&FilmId=<%FilmId%>&Ref=<%ImageRef2%>`
	
	The field
	
	`<%ImageRef2%>`
	
	contains the Reference code for the second image, and so on.

<%Rating%>	The age rating.
<%ReleaseDate%>	The release date. If no value is present, this will be empty.
<%Reviews%>	A link to reviews.
<%RunningTime%>	The running time for the film.
<%\Title%>	The title of the film.

<%WideReleaseDate%>	The nationwide release date, as read from the tvmv.dbo.FilmInfo table.

MVMovieInfoImageItem.pbo

<%Caption%>	The caption for the image.
<%FilmId%>	The ID for the film. It is used in the zoom link.
<%ImageLink%>	The URL for the image.
<%ImageRef%>	The reference code for the image. It is used in the zoom link like this: `<%pbcs.dll%>/misc?url=/templates/zoom.pbs&FilmId=<%FilmId%>&Ref=<%ImageRef%>`

MVMovieInfoImageRow.pbo

This file is output by the field

```
<%images%>
```

in MVmovieInfo.pbo. The file is output for each row of images. It will be output for as many rows as needed.

<%Col1%>	Outputs the image for the first column.You also have `<%Col2%>, <%Col3%>` and so on up to the number of columns you specified in the tag.

MVMovieInfoReviewItem.pbo

This field is output for each review.

<%Body%>	The body of the review
<%Description%>	A description of the reviewer.
<%FilmID%>	The film ID.
<%\Title%>	The title for the review.

pbs mv_schedule

<pbs:mv_schedule>

The tags can be placed on all templates. The example uses **mv_ovr.pbs**.

The tag is used like this:

```
<pbs:!mv_schedule ZipCode="43660" useobjects=1 width="625">
```

Options

Radius	Sets the search radius in miles.
Timeformat	Please note that it is not possible to include the date in **Timeformat.** I t outputs the time and should be used as in this example: `<pbs:!mv_schedule zipcode="9000" useobjects=1 width="625" timeformat="H:nn:ss AM/PM">`
Useobjects	Enables the use of object files.
Width	Sets the width of the output.
Zipcode	Sets the default zip code for the listing.

MVSchedule.pbo

<%ScheduleContent%>	Outputs the content.

MVScheduleChainFooter.pbo

This file has no fields.

MVScheduleChainHeader.pbo

<%ChainName%>	The name of the movie theater chain.

MVScheduleMovieItem.pbo

<%FilmID%>	The ID for the film.
<%MovieName%>	The name of the movie.
<%Rating%>	The movie's rating.
<%RunningTime%>	The running time.
<%ShowTimes%>	The show times.
<%TheaterID%>	The ID for the movie theater that is sent in the feed to the TVMV database.

MVScheduleTheaterFooter.pbo

<%Items%>	Outputs all the movie items.

MVScheduleTheaterHeader.pbo

<%Address%>	The address.
<%City%>	The city name.

<%Phone%>	The phone number.
<%State%>	The state.
<%TheaterName%>	The name of the movie theater.
<%ZipCode%>	The zip code.

Search form for movie guide

The form is set up like this:

```
<form name="moviesearchform" action="<pbs:prog>/section?Category=MV" method="POST">
Fields and buttons go here.
</form>
```

You must set the category in the action field to the category in which the movie listing will be displayed.

Fields and buttons

Date field	Used to select the date for which movie listings will be shown. Use the Getoptions tag to get a list of dates currently in the database.
	`<select name="Date">`
	`</select>`

Radius	`<input name="Radius"` `value="<pbs:!#Radius#>">`
Search button	`<input  type="button" value="Go!"` `onclick="javascript:DoSubmit();">`

Zipcode	Used to enter the zip code for which movie listings will be shown. The zipcode tag is used to extract a zip code the web guest has previously entered.
	`<input name="ZipCode"`
	`value="<pbs:!#ZipCode#>">`

The JavaScripts included in the examples above are required for the search form to work.

The form can use the following tags:

Get options	Extracts options from the database. To extract available dates, use the tag like this: `<pbs:tv_getoptions type="MVDates" dateformat="dddd, mmmm d">` The **Dateformat** option is used to set the format of the dates shown in the drop-down.
Radius	Extracts a previously entered search radius. The tag is used like this: `<pbs:!#Radius#>`
Zipcode	Extracts a previously entered zip code. The tag is used like this: `<pbs:!#ZipCode#>`

Other Tags

This chapter describes miscellaneous tags that are used with SAXOTECH Online.

- pbs calendar
  - Calendar.pbo
  - CalendarDateItem.pbo
  - CalendarWeekItem.pbo
- pbs Captcha
  - Captcha.pbo
- pbs category
- pbs categorydescription
- pbs categoryid
- pbs catselect
- pbs datecalc
- pbs datechoice
- pbs delimiter
  - Delimiter.pbo
- pbs dingbats
  - DingBats.pbo
- pbs frontpagearchive
  - FPArchive.pbo
  - FPArchiveItem.pbo
- pbs getcookie
- pbs getheaderinfo
  - HeaderInfo.pbo
- pbs GetKeywords
  - KeywordList.pbo
- pbs lastupdated
- pbs link
- pbs metarefresh
- pbs parvariables name="site"
- pbs PbcsRequestRandomA
- pbs PbcsRequestRandomB
- pbs profileid
- pbs profilename
  - ProfileName.pbo
- pbs prog
- pbs query
  - CustomQuery_queryname.pbo
  - CustomQuery_querynameitem.pbo
- pbs queryparam
- pbs random
- pbs script
- pbs sessioninfo
- pbs setheaderinfo
- pbs showdate
- pbs statuslog
- pbs taxonomyinfo
  - TaxonomyInfo.pbo
  - TaxonomyInfoItem.pbo
  - TaxonomyInfoItemPathItem.pbo
- pbs UniquePageID
- Pop-up calendar
- sessioninfo with static cache files

pbs calendar

<pbs:calendar>

This tag outputs a calendar that you can use to browse your content. This tag can be used on all templates.

Options

Eventdates	Determines whether only dates with events should be active. Enter `eventdates=1` to enable this option.
Futuredates	Determines whether the calendar should show future dates. Enter `futuredates=0` to disable this option.
Objectclass	Sets the object class for the calendar.
Pastdates	Determines whether the calendar should show past dates. Enter `pastdates=0` to disable this option.
Type **From version 6.6.0sr36, 6.7.0sr8**	Sets the event resource. `Type=cce` defaults to agenda
Weekstart	Sets the first day of the week. 1 is Sunday, 2 is Monday, and so on.

URL Options

CalDate	Determines which month the calendar should show.

Calendar.pbo

<%CalChangeScript%>	Outputs a script that can be used with the previous and next month fields to browse the calendar. It can be used like this:
	`<a href="javascript:calChange('<%PrevCalDate%>')">Previous</a>`
	`<a href="javascript:calChange('<%NextCalDate%>')">Next</a>`

<%MonthName%>	The name of the current month.
<%NextCalDate%>	The next month in YYYYMM format.
<%NextMonthName%>	The name of the next month.
<%NextShortMonthName%>	The name of the next month in short form.
<%PrevCalDate%>	The previous month in YYYYMM format.

<%PrevMonthName%>	The name of the previous month.
<%PrevShortMonthName%>	The name of the previous month in short form.
<%ShortMonthName%>	The name of the current month in short form.
<%WeekItems%>	Lists the week items.
<%Year%>	The year.

CalendarDateItem.pbo

<%Active%>	This field has the value 1 if the date is active. It is speculated that this variable will have a value of 1 if there is an event that occurs on the date currently being output (need confirmation from Development).
<%Date%>	The date.
<%FirstInWeek%>	This field has the value 1 if the day is the first in the week.
<%ISODate%>	The date in ISO format. It can be used like this: `<a href="events?Category=EVENTS&Date=<%ISODate%>&CalDate=<%ISODate%>">`
<%LastInWeek%>	This field has the value 1 if the day is the last in the week.
<%NotInMonth%>	This field has the value 1 if the day is not in the current month.
<%Today%>	This field has the value 1 if it is today's date.

CalendarWeekItem.pbo

<%CurrentWeek%>	This field has the value 1 if it is the current week.
<%DateItems%>	Lists all the days in the week.
<%WeekNumber%>	Contains the week number.

pbs Captcha

<pbs:captcha>

<pbs:captcha> most run without cache. We recommend what you run this pbs tags in a Ajax call with a misc action. (misc is not cached with our FastCache level.)$('#Newsfeed_bottom').load("/apps/pbcs.dll/misc?url=/misc/captcha.pbs&omniture=0");

The <pbs:captcha> tag represents a visual captcha. Currently, the tag can be used everywhere, but server side validation is only implemented for forums and xsendmail for now.

The tag supports the following options:

Length	The number of characters to render in the captcha. Default is 6 characters.
BackgroundColor	Sets the color of the captcha background, use hex RGB value.
FontColor	Sets the color of the captcha font, use hex RGB value.
NoiseColor	Sets the color of the captcha noise, use hex RGB value.
ObjectClass	Specifies what object file design class to use.

If there are no color values, or only some values set on the tag, it looks for any corresponding values in the publicus.ini. If there are no values at all, it defaults to the original color scheme (white/blue/light blue). Faulty written color values return black as the default color.

```
<pbs:!captcha length=N>
```

```
<pbs:!captcha Length=5 FontColor=RRGGBB NoiseColor=RRGGBB BackgroundColor=RRGGBB
ObjectClass=200>
```

The tag outputs the content of its object file, Captcha.pbo, specifically the file Captcha200.pbo.

Check a user's solution

Checking if a solution is correct can be done in the pbcs.dll by using the following action:

```
<%pbcs.dll%>/checkcaptcha?CaptchaID=X&Captcha=Y&Unpersist=Z
```

CaptchaID	The <%ID%> in captcha object file.
Captcha	The user's solution to the captcha.
Unpersist	Whether or not to remove the captcha from the user's session file. Default is 0. The unpersist can be forced with the **ForceUnpersist** setting in the CAPTCHA section

The action returns an HTTP response of **200 OK** if the check is successful and **400 BAD REQUEST** if the check fails.

Captcha.pbo

Captcha.pbo supports the following variables:

<%ID%>	The ID of the captcha.
<%image%>	The image URL to the visual captcha.
<%length%>	The number of characters rendered.

In a case where the <%image%> variable is empty, check the following settings in publicus.ini:

```
[GENEREL]
Domain=mydomain.com
ImageServer=myimagedomain.com
[Image]
Server=myimagedomain.com
```

It is important that at least one of these settings is fulfilled. If not, the captcha image will be blank.

An example object file (integrated into a form constructed as a simple name-value table:

```
[%{image}
  <tr>
    <td width="22%">
      <b>Captcha</b>
    </td>
    <td width="78%" valign="middle">
      <input type="hidden" name="CaptchaID" value="<%ID%>">
      <img src="<%Image%>" border="0" title="Captcha"> <input type="text"
name="Captcha" maxlength="<%Length#+5%>" size="<%Length#+3%>">
    </td>
  </tr>
%]
```

pbs category

<pbs:category>

This tag outputs the name of the category currently being viewed.

Options

Category	Makes the tag output the name of the category specified, for example: `<pbs:category category=NEWS>`
Main	Makes the tag output the name of the main category when viewing a subcategory.
Section	Makes the tag output the section category instead of the story's category. The section category is inherited from the section front on which the story is displayed. If the section category is blank, this option has no effect.

By using the tag as follows:

```
<pbs:category section main>
```

you will get the name of the maincategory for the sectioncat you are viewing.

pbs categorydescription

<pbs:categorydescription>

This tag outputs the category description or information. The description is stored in the database. Without options, the category description field is output.

Options

Category	Makes the tag output the category.
Info	By default, this option is not used. Enter information in the tag as an option to output the info field instead of the description field.

Example Usage

```
<!-- display the current category description -->
<pbs:categorydescription category>

<!-- display the current category info -->
<pbs:categorydescription info>
```

pbs categoryid

<pbs:categoryid>

This tag outputs the name of the category ID currently being viewed. This function can be used to include an image based on the category. It can be used like this:

```
<img src="/image/header_<pbs:categoryid>">
```

Because the include tags are parsed first, the tag **cannot** be used in the include tag like this:

```
<!--#include virtual="/includes/<pbs:categoryid>.inc"-->
```

Options

Main	Makes the tag output the ID of the main category when a subcategory is being viewed.
Section	Makes the tag output the section category instead of the story's category. The section category is inherited from the section front on which the story is displayed. If section category is blank, this option has no effect.

pbs catselect

<pbs:catselect>

Enables you to output text based on the category being viewed. This tag can be used in all templates.

The tag might appear as follows:

```
<pbs:catselect selection="frontpage,news" values="#3265CB,#003399" default="#3265CB">
```

Options

Default	Sets the default value.
Selection	Used for the list of categories.
Values	Used for the list of values for each category in the selection option.

pbs datecalc

<pbs:datecalc>

The tag for date calculation is called `<pbs:datecalc>`. It allows you to do date calculations for search forms and other fields.

Without any options, the tag will output the active publication date.

Options

Dateformat	This tag supports the SAXOTECH Online Date format for output. Date format is documented in the section Formatting dates.
Days	Sets the offset from today's date in days. By default, this is not set. The days can be set to any value. This is added onto the publication date. You can also set days=-4 to subtract four days from the publication date.
Firstofmonth	Sets the date to the first day of the month.
Firstofweek	Sets the date to the first day of the week.
Lastofmonth	Sets the date to the last day of the month.
Lastofweek	Sets the date to the last day of the week.
Lastofyear	Sets the date to the last day of the year.
Firstofyear	Sets the date to the first day of the year.
Months	Sets the offset from today's date in months. By default, this is not used. The months can be set to any value. This is added onto the publication date. You can also set months=-4 to subtract four months from the publication date.
Now	Outputs today's date.
URL	Makes the tag use the date in the URL as the start date for the calculation.
UTC	Set GMT Time (UTC time)
Weekstart	Determines whether Sunday or Monday is the first day of the week. Setting this option to 0 makes Sunday the first day of the week. Setting it to 1 makes Monday the first day of the week. 0 is the default. You can set Weekstart to be any number from 0 to 6.

The tag is most often used in search tags to set date ranges. To make a form that searches this month's stories, use:

```
<pbs:datecalc firstofmonth>
```

in the **from date** field, and

```
<pbs:datecalc lastofmonth>
```

in the **to date** field.

To search the last 7 days of stories, use:

```
<pbs:datecalc days=-7>
```

in the **from date** field, and

```
<pbs:datecalc now>
```

in the **to date** field.

To search only the previous month's stories, use:

```
<pbs:datecalc firstofmonth days=-2 firstofmonth>
```

in the **from date** field, and

```
<pbs:datecalc firstofmonth days=-2 lastofmonth>
```

in the **to date** field.

pbs datechoice

<pbs:datechoice>

This tag is used to create a list of date options for a drop-down menu. It always uses todayÕs date.

Options

Dateformat	Sets the format for the date that is displayed.
Days	Sets the number of days for which the system should create options.

The tag

```
<pbs:datechoice days=3 dateformat="mm/dd/yyyy">
```

will output this:

```
<option value="20030806" selected>Today</option>
```

```
<option value="20030805">08/05/2003</option>
```

```
<option value="20030804">08/04/2003</option>
```

pbs delimiter

<pbs:delimiter>

This function can be used everywhere to output a delimiter.

Delimiter.pbo

This file has no fields.

pbs dingbats

<pbs:dingbats>

This function is used to output special characters. It can be used everywhere.

Options

Char	Determines which character should be shown. You use the option like this: `<pbs:dingbats char="n">`

DingBats.pbo

<%Character%>	This field contains the value from the char option.

An example of the object file output would be:

```
<img src="/images/dingbats/<%Character%>.gif" border=0>
```

pbs frontpagearchive

<pbs:frontpagearchive>

This tag makes links to the front pages for the last week. This tag can be placed on all templates. If you are going to use this option, your **Newslis t** tags on the front page must contain the option `ado=1`.

The tag might appear as follows:

```
<pbs:frontpagearchive days=14  sort=desc dateformat="dddd, mmmm dd">
```

Options

Dateformat	Used to set the date format. For more information, please see Format ting dates.
Days	Determines the number of days for which the tag should output links. By default, links are output for 6 days. This option can be set from 1 to 6 days.
Sort	Used to sort the list of dates. By default, days are sorted in ascending order. Enter `sort=desc` to sort the days in descending order.

Weekstart	Used to show the front pages from yesterday to the start of the week. By default, this option is not used. To use this option, set the option to the first day of the week.

This tag will use object files if they exist.

FPArchive.pbo

<%FPArchiveItems%>	Contains all links.

FPArchiveItem.pbo

<%\Date%>	The date in the formatted format.
<%ISODate%>	The date in ISO format.
<%LastItem%>	Contains a space for the last item. This field can be used for extra spacing after the last record.

pbs getcookie

<pbs:getcookie>

This tag is used to read a cookie. Use a script to set a cookie. This tag can be used on all templates.

Options

Name	This option specifies the name of the cookie you want to read. By default, this option is not set. Enter `name="XXX"` where XXX is the name of the cookie.

pbs getheaderinfo

<pbs:getheaderinfo>

This tag is used to read HTTP headers.

Options

ExtendedHeaders	Used to specify extra headers to be read. By default, this option is not used. Any headers not included in the HeaderInfo.pbo variable list must first be defined here. For example, entering: `<pbs:getheaderinfo ExtendedHeaders="Accept-Encoding">` will allow access to a variable named <%Accept-Encoding%> in the HeaderInfo.pbo.

HeaderInfo.pbo

These fields are derived form the message headers defined in the standard HTTP protocol. Any headers not included in this list must first be pre-defined in the pbs getheaderinfo tag using the ExtendedHeaders parameter.

<%Accept%>	Used to specify certain media types, which are acceptable for the response. Example: `Accept=text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5`
<%Authorization%>	Consists of credentials containing the authentication information of the client for the realm of the resource being requested.
<%CacheControl%>	The cache control settings sent by the browser. Example: `CacheControl=no-cache`
<%Connection%>	Specifies options that are desired for the particular connection. This must not be communicated by proxies over further connections. Example: `Connection=keep-alive`
<%Content%>	The entity-body that is sent. For example, names and values from a form. Example: `Content=SearchTitle=Cars&SearchBody=Volvo&SearchBodyTitle=Brand&SearchRegionID=&SearchSubjectID=`
<%ContentEncoding%>	Specifies which data format transformations (called content (en)codings), such as compression mechanisms, are acceptable for the response, and assigns preferences to them.
<%ContentType%>	Specifies the Internet media type of the entity-body that is sent or would have been sent if requested. Often includes a **char set** parameter to specify the character encoding. Example: `ContentType=application/x-www-form-urlencodedHost=localhost`
<%ContentVersion%>	Defines the version tag associated with a rendition of an evolving entity. Combined with the **Derived-From** field, it allows a group of people to work simultaneously on the creation of a work as an iterative process. The field should be used to allow evolution of a particular work along a single path. It should not be used to indicate derived works or renditions in different representations. It may also be used as an opaque value to compare a cached entity's version with that of the current resource.
<%DerivedFrom%>	When an edited object is resubmitted (for example, using PUT), this field gives the value of the Version . This typically allows a server to check that two concurrent modifications by different parties will not be lost, and to use established version management techniques to merge both modifications.

<%Host%>	Specifies the Internet host of the resource being requested. Example: `Host=localhost`
<%InternalPathInfo%>	Contains the path to the page being shown. Example: `InternalPathInfo=/frontpage`
<%InternalScriptName%>	Contains the path to the SAXOTECH Online dll. Example: `InternalScriptName=/apps/pbcs.dll`
<%Method%>	The method for accessing the page. Example: `Method=GET`
<%PathInfo%>	Additional path information, as given by the client. This consists of the trailing part of the URL, after the script name but before the query string, if any. Example: `PathInfo=/frontpage`
<%PathTranslated%>	The value of **PathInfo,** but with any virtual path name expanded into a directory specification. Example: `PathTranslated=c:\inetpub\wwwroot\frontpage`
<%ProtocolVersion%>	Gives the version of the information retrieval protocol relating to this request. Example: `ProtocolVersion=HTTP/1.1`
<%Referer%>	Used by a client to specify the address (URL) of the resource from which the Request-URL was obtained. This is for the server's benefit. Example: `Referer=http://localhost/apps/pbcs.dll/frontpage?nocache=1`

<%RemoteAddr%>	The IP address of the client or agent of the client (for example, gateway or firewall) that sent the request. Example: `RemoteAddr=127.0.0.1`
<%RemoteHost%>	The host name of the client or agent of the client (for example, gateway or firewall) that sent the request. If the server does not have this information, it should set **RemoteAddr** to leave this unset.
<%ScriptName%>	A virtual path to the script being executed, used for self-referencing URLs. Example: `ScriptName=/apps/pbcs.dll`
<%Title%>	The title of the document. Not part of the document. The syntax is the same as for the **Subject** field in RFC822. The **Title** field differs from the **Subject** field in that the title is defined by the author of a document, but the **Subject** field is defined by the originator.
<%URL%>	Gives the base portion of the URL. Example: `URL=/apps/pbcs.dll`
<%UserAgent%>	Contains information about the user agent (client) originating the request. Example: `UserAgent=Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7) Gecko/20060909 Firefox/1.5.0.7`

pbs GetKeywords

<pbs:GetKeywords>

This tag extracts keywords from stories. It is typically used on a search page to populate a drop-down. This tag can be used on all templates.

Options

Categories	Use this option to extract keywords from several categories. Example: `categories="NEWS%,SPORTS%"`

Category	Sets the category from which the keywords will be extracted. Example: `category="NEWS"`
Objectclass	Sets the object class for the object file.
Options	When `options=1` is used, the object file is not used and the tag will output `<option>` tags for a `<select>` list.

The tag is used to output options for the **keyword** field in a search form:

```
<select name="Keyword" size="1">
<option value=" " SELECTED>Select Topic</option>
<pbs:getkeywords category="NOTICES" options=1>
</select>
```

KeywordList.pbo

<%Keyword%>	Outputs the keyword.

pbs lastupdated

<pbs:lastupdated>

Used to show the date on which a story was last updated. This tag can be used on all templates.

The tag is used like this:

```
<pbs:lastupdated Category="Sports" format="ddd dd.mm">
```

Options

Format	Sets the date format. The legal values are described in the section Formatting dates.
Category	Determines the category for which the date will be shown. Category has the following options for use: • Category="%"* Shows the last updated date for all categories. • Category="NEWS"* Shows the date stories in the NEWS category were last updated. • Category="NEWS01,NEWS02"* Shows the date stories in the NEWS01 or NEWS02 categories were last updated. • Category="NEWS%"* Shows the date stories in all of the NEWS categories were last updated.
Lower	Makes the system output the text in lowercase.
Norwproper	Makes the system output the text with the first letter in the first word in uppercase.
Object	Makes the system output the lastupdated tag based on the specified object file (Please note that if this option is used, the other format settings are disregarded.): • Ex: object=OBJECTFILENAME

Proper	Makes the system output the text with the first letter in each word in uppercase.
Upper	Makes the system output the text in uppercase.

Object File Variables

<%IsoDate%>	Outputs the date in standard ISO format "yyyymmddhhmmss". This allows slightly more customization than the tag's "Format" option.
<%Date%>	Outputs the date in "dd/mm/yyyy" format.

pbs link

<pbs:link>

```
<pbs:link article inherit template=printart text="Printer-friendly version">
<pbs:link>
<pbs:link inherit ondate=0 href text=img:>&MaxW=580">
```

pbs metarefresh

<pbs:metarefresh>

This tag is put in the `<head></head>` section to enable pages to auto refresh. This tag can only be used on front pages, section fronts, and stories. The {{<pbs:metarefresh> }}tag has only one option, refresh time. The refresh time is set in the **web.siteconfig** table. You can set separate refresh times for the front page, section front, and story. The default refresh time is 10 minutes.

pbs parvariables name="site"

pbs:#parvariables# name="site"

It is currently not possible to use all story parameters as global variables. This tag can be used to retrieve parameters:

```
pbs:#parvariables# name="site"
```

Where name can be set to the following values:

site	The site indicated in the publicus.ini.
date	The date in the URL. If there is no date, the current date will be used.
category	The category in the URL.
storyid	The story ID in the URL. This is only available on the /article action.
publishdate	The current site publish date.
ref	The story or photo gallery AR/PH.
theme	The theme from the URL.
keywords	The keywords from the URL.
profile	The profile ID from the URL.
profilestoryguid **From versions 6.9.0sr27, 7.0.0sr2**	Contains the GUID of the default story of the context profile.
sectioncat	The section category from the URL.
lightid	The light ID from the URL.

pbs PbcsRequestRandomA

<pbs:#@PbcsRequestRandomA#>

This tag is processed when using static cache files. It is a random number between 0 and 1000000.
To enable this tag for post processing make the following publicus.ini setting:

```
[CACHE]
EnablePostProcessRandomNumbers=1
```

pbs PbcsRequestRandomB

<pbs:#@PbcsRequestRandomB#>

This tag is processed when using static cache files. It is a random number between 0 and 1000000.
To enable this tag for post processing make the following publicus.ini setting:

```
[CACHE]
EnablePostProcessRandomNumbers=1
```

pbs profileid

<pbs:profileid>

This option gives you the current ID for the profile. This tag works in the same way as pbs categoryid.

pbs profilename

<pbs:profilename>

This tag extracts the profile name on a story.

Options

ID	Specifies which ID the tag should show. By default, it shows the current profile. Enter `id=x` to show the name for profile x.
Objectclass	Sets the object class.

ProfileName.pbo

<%IsCurrent%>	This field has the value 1 if the profile is the current profile.
<%ProfileID%>	The profile ID.
<%ProfileName%>	The profile name.

pbs prog

<pbs:prog>

To output the path to the SAXOTECH Online dll, you have two functions. These functions should be used instead of writing the path directly in the document.

In pbs files, you can use the function <pbs:prog> to output the path to the SAXOTECH Online dll.

In pbo files you can use the function <%pbcs.dll%> to output the path to the SAXOTECH Online dll.

pbs query

<pbs:query>

The tag is used to execute a stored procedure that you have written and output the content as part of the page. The tag is written as follows:

```
<pbs:query name="Queryname">
```

It will look for the stored procedure in the database. You must must have direct access to the database so you can execute queries and create stored procedures. Custom queries can only use stored procedures. In the stored procedure you can reference:

- Stored procedures
- Tables
- Functions

All references to functions, stored procedures, and tables require full database and owner qualifiers, for example, **custom.dbo.mystoredprocedure**. You can also make references outside the custom database. When doing this you also need full qualifiers, such as **web.dbo.brukere**. In addition, all selects should use the **(nolock)** hint to ignore locks that might slow down the query.

The tag has the following options:

Objectclass	This option sets the object class for the object files. This option is not required.
Name	Name sets the name of the stored procedure. Name is required. The name must contain only the following characters: • Letters from A to Z (both upper case and lower case) • Numbers from 0 to 9 • Underscore (_)
Dump	`dump=1` Log query informations

Parameters	This option is used for sending parameters to the query you have written. This option is not required. The parameters are sent from the tag must match the parameters the query requires. Otherwise, the query will fail. You can use @ to specify a specific parameter name. All parameters need values; stored procedures may specify defaults.The parameters should be passed using the syntax:
	`"parametername"="parametervalue"`
	So the query tag might appear as follows:
	`<pbs:query Name="foo" parameters="@date=6/20/2008,@count=<%count%>,@startdate=<%start%>">`

In this example, we use the parameters option to pass three parameters to the query in the order specified:

```
<pbs:query Name="foo"
parameters="@date=6/20/2008,@count=<%count%>,@startdate=<%start%>">
```

In this example, two of the parameters are sent as blank:

```
<pbs:query Name="foo" parameters="@date=6/20/2008,,">
```

In this example, the query is inside a macro accepting fields. The count parameter is skipped, so **@start=** is used to specify the last parameter:

```
<pbs:query Name="foo" parameters="@date=6/20/2008,@startdate=<%start%>">
```

If you send in more or less variables than the system expects, the system will expose fully qualified SQL server messages on the results page. Always use parameter names. This makes a longer string but makes no difference in performance. Sending all parameters makes the tag easier to read and follow later.

The tag can be placed directly in a template. The tag must then be run cached. If you need to run the tag with recache intervals or you need to pass in variables, the tag must be placed inside a macro or an <pbs:include>.

For more information and examples of custom queries, please see the Queries section of the Online User Manual.

CustomQuery_queryname.pbo

Replace **queryname** with the name of the query you are running.

<%items%>	This field outputs all items.

CustomQuery_querynameitem.pbo

Replace **queryname** with the name of the query you are running. All column names become fields in the object file. In addition, the following fields are available:

<%counter%>	This field contains the counter value for the record shown.
<%islastitem%>	This field has the value 1 on the last record.
<%odd%>	The field has the value 1 if the record number is odd.
<%even%>	The field has the value 1 if the record number is even.

pbs queryparam

<pbs:queryparam>

This tag outputs query parameters on the page. A query parameter is the part of the URL that comes after the ? character. This function is usually used to put values from the link in forms for further use.

Options

Param	Used to select the parameter that will be output.

The following code will output the category ID for the category being viewed:

```
<pbs:queryparam param=category>
```

pbs random

<pbs:random>

This tag is used to create a variable with a random number. This number can be used in other functions.

Options

ID	Sets the name for the global variable. If the ID is set to xxx, the value will be called `<%@RandomValue_xxx%>`
Max	Sets the maximum value.
Min	Sets the minimum value.

This example shows how you can output different tags based on the random value.

The tag might appear as follows:

```
<pbs:include virtual="/includes/ads.inc" cachetime=2>
```

The Ads.inc might appear as follows:

```
<pbs:random min=1 max=3 id=adselect>
<pbs:macro name=adselect>
```

The AdSelect.pbo might appear as follows:

```
[%{@RandomValue_adselect=1}
<pbs:clasifieds......jobs>
%]
[%{@RandomValue_adselect=2}
<pbs:clasifieds......houses>
%]
[%{@RandomValue_adselect=3}
<pbs:clasifieds......autos>
%]
```

pbs script

<pbs:script>

The script tag is used to execute scripts written in other languages from within the SAXOTECH Online publishing platform. This function uses the Windows Scripting Host. By default, VBScript and JavaScript are supported. The SAXOTECH Online publishing platform supports any language supported by the Windows Scripting Host, including Perl, PHP, Python, .NET, and so on.

Saxotech strongly recommends that prior to using the script tag you receive additional training and read the SAXOTECH Online Scripting Technical User Manual.

All scripts must be placed in the **scripts** folder in the **templates** folder. The typical location is **/templates/scripts.**

The basic tag looks like this:

```
<pbs:script name="scriptname" language="Vbscript">
```

Options

Name	Sets the name of the script.

Language	The language sets the script language used. By default, only VBScript and JavaScript are installed on the web server. If you want to use additional script languages, you must install support for the languages on the server before they can be used. The language used in the file must match the language specified in the tag. Examples:
	
	`<pbs:script name="scriptname" language="VBScript">`
	
	The system will look for the file **scriptname.vbs.**
	
	`<pbs:script name="scriptname" language="Jscript">`
	
	The system will look for the file **scriptname.js.**
	
	`<pbs:script name="scriptname" language="PerlScript">`
	
	The system will look for the file **scriptname.pl.**
	
	`<pbs:script name="scriptname" language="PHPScript">`
	
	The system will look for the file **scriptname.php5.**
	
	`<pbs:script name="scriptname" language="ActivePHP">`
	
	The system will look for the file **scriptname.phs *or*** scriptname.php.*
	
	`<pbs:script name="scriptname" language="Python">`
	
	The system will look for the file **scriptname.py.**
	
	`<pbs:script name="scriptname" language="RubyScript">`
	
	The system will look for the file **scriptname.rb.** All other languages use the extension `.wsf` for the script file.

ShowError **From version 6.9.0sr12**	Makes it possible to suppress all errors generated by the script. `<pbs:script name="scriptname" language="RubyScript" showerror="0">` Default is show errors.
Cache time	The tag does not support the cache time parameter. You must place the tag inside a macro or an include file.

Variable name	It is possible to parse information to the script, by creating new variables in the scripting file, example:
	
	`gen_ovr.pbs:`
	`<pbs:script name="helloworld.js"mystring="Hello" mystring2="you">`
	
	`/templates/scripts/helloworld.js:`
	`SOSE.ECHO(SOSE.GETVAR("mystring") + " " + SOSE.GETVAR("mystring2"));`
	
	Notice that the name of the variable can't be the same as the default options of the tag.

Variables can be defined on the fly in the tag using the same method as the `<pbs:macro>` tag.

The $S modifier can be used on variables in any object file that is in the SAXOTECH Online Tags Manual. This modifier is used to call a script. Note that you cannot use memory banks when using the $S modifier.

The SAXOTECH Online object is used to communicate with Online from inside the script. For the complete list of properties and methods available for the SAXOTECH Online object, please see the SAXOTECH Online Scripting Technical User Manual.

pbs sessioninfo

<pbs:sessioninfo>

With pbs:sessioninfo it is possible to get session information about your site visitors. When an anonymous user visits your website, pbs:sessioninfo can provide information like which user agent or IP address the user came from.

pbs:sessioninfo can also be used with Access Control (used to limit access to certain content), so when a user logs in pbs:sessioninfo can output registered user information like first name, last name and email address.

Variables and object file

A full list of all available session information in SAXOTECH Online can be found here: Session file sections and settings.

Session file sections and settingsThe object file is called SessionInfo.pbo, and is located in the *Objects* folder.

All available session data can be output in the object file by entering <%FirstName%>, <%LastName%>, <%EmailAdr%> exactly as in the session name.

Example

pbs:sessioninfo can be used to check if the user is logged in or not:

```
[%{LoginName}
<!-- Show user registered information. -->
Welcome <%LastName%>, <%FirstName%>
%|%
<!-- The user is not logged in. Show login form -->
<form><input type="text" .....></form>
%]
```

Caching

Always use <pbs:#sessioninfo#> when using static full page caching to avoid registered user information being cached in the cache file and presented to other users. Read more about sessioninfo with static cache files.

Parameters

<pbs:#sessioninfo#> takes no parameters.
<pbs:sessioninfo> supports objectclass="x".

pbs setheaderinfo

<pbs:setheaderinfo>

This tag is used to change values of the HTTP header fields available through pbs getheaderinfo, either as the standard HTTP header fields or fields defined through the pbs getheaderinfo parameter.

The

```
<pbs:setheaderinfo>
```

If you are using full page cache remember to use this pbs tag instead:

```
<pbs:#setheaderinfo#>
```

tag is used by adding parameters, of the form Name=value. This will add a new header or overwrite an existing one of the same name.

Example 1	`<pbs:setheaderinfo HeaderName="HeaderData" ... >`
Example 2	`<pbs:setheaderinfo Accept-Encoding="Yes" ... >`

Example 3	```<pbs:setheaderinfo remotehost=newXPhost> <pbs:getheaderinfo>```
	If your HeaderInfo.pbo file is like the following:
	``<%remoteaddr%>
	 <%remotehost%>```
	Then where the output was previously `Gandalf192.168.172.1`It will now be `newXPhost192.168.172.1` ||

Example 4	<pbs:#setheaderinfo# Cache-Control="no-cache, no-store, max-age=0, s-max-age=0, must-revalidate, proxy-revalidate">
Example 5	<pbs:!setheaderinfo Cache-Control="no-cache, no-store, max-age=0, s-max-age=0, must-revalidate, proxy-revalidate">
Example 6	<pbs:setheaderinfo Content-type=text/xml;charset=iso-8859-1>

pbs showdate

<pbs:showdate>

This function allows you to output a date. This tag can be used on all templates.

The tag is used like this:

```
<pbs:showdate format="ddd dd.mm" offset=17>
```

Options

Format	Sets the date format. The legal values are described in the section Formatting dates.
Lower	Makes the system output the text in lowercase.
Norwproper	Makes the system output the text with the first letter in the first word in uppercase.
Offset	Determines how many days the date is offset from today's date.
Proper	Makes the system output the text with the first letter in each word in uppercase.
Upper	Makes the system output the text in uppercase.

pbs statuslog

<pbs:#statuslog#>

The tag `<pbs:#statuslog#>` is a debugging function that outputs the build time of the page. This tag should be included commented out at the bottom of your templates so you can easily check the build time for the page.

You can read more about <pbs:#statuslog#> and learn how to use it in the section Testing.

pbs taxonomyinfo

<pbs:taxonomyinfo>

The following tag can be used to retrieve taxonomy information based on taxonomy IDs:

```
<pbs:taxonomyinfo [taxonomyids|taxids]="xxx,yyyy,zzzz" objectclass="qqq"
delimiter="AChar">
```

Options:

TaxonomyIDs/TaxIDs	Specifies which ID the tag should show.
Objectclass	Sets the objectclass for the files

Delimiter	A character to use to separate taxonomy words if more than one is specified

If there is a query parameter called taxonomyids or taxids then the tag parameter is ignored.

The following object files are used to generate the content of this tag:
TaxonomyInfo.pbo
TaxonomyInfoItem.pbo
TaxonomyInfoItemPathItem.pbo

TaxonomyInfo.pbo

This file contains the main framework for the tag. It is possible to use the condition tags to exclude the whole file from being shown.

<%items%>	This tag outputs all the items.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	Outputs the total number of taxonomywords in the list.

TaxonomyInfoItem.pbo

The following list of parameters is available for use in this object file:

<%id%>	The ID of the taxonomy word.
<%description%>	The description of the taxonomy word.
<%url%>	The URL associated with the taxonomy word.
<%path%>	The parent tree of the taxonomy word. The delimiter parameter is used to separate items in the tree. Please note that for delimiter "\" you have to use the HTML value \
<%PathItems%>	Uses ArticleTaxonomyItemPathItem.pbo
<%Counter%> **From versions 7.1.0sr13, 7.2.0sr2**	The taxonomy word's number in the list.
<%TotalCount%> **From versions 7.1.0sr13, 7.2.0sr2**	The total number of taxonomy words in the list.
<%IsLast%> **From versions 7.1.0sr13, 7.2.0sr2**	Has the value of 1 if the taxonomy word is the last one in the list.

TaxonomyInfoItemPathItem.pbo

Builds the tree from its top parent

<%parent%>	The description of the parent taxonomy word
<%islast%>	1 if the item is the last in the tree
<%isfirst%>	1 if the item is the first in the tree

pbs UniquePageID

<pbs:#@UniquePageID#>

Performs a lookup in the localdata.ini file to find a mapping for the following actions:

- /frontpage (default value frontpage)
- /CCE (default value CCE+Module)
- /section (default value category)

To enable this tag for post processing make the following publicus.ini setting:

```
[CACHE]
EnablePostProcessRandomNumbers=1
```

Pop-up calendar

The pop-up calendar is used when you want the web guest to be able to pick a date from a calendar. This function is typically used for the events listing.

The following JavaScript is needed to open the calendar:

```
<script language="javascript">
<!--
function choosedate () {
 window.open('/g/kalender_eng.html?
formname.fieldname&nobackdate'
,'','menubar=0,titlebar=0,width=268,height=236');
 document.searchform.Interval.options[document.searchform.Interval.options.length-1].s
elected = true;
}
function ResetDate () {
 if (document.searchform.Interval.selectedIndex !=
(document.searchform.Interval.options.length-1))
 {
  document.searchform.dateselected.value = "";
 }
}
-->
</script>
```

Options

Formname.fieldname	Used to select the field in which the date will be placed. Both the form name and the field name must be specified.
Limit	Used to set the limit for the dates that can be picked. For example: `&limit=20010101-20011231` The option can be used with the pbs datecalc tag.
Nobackdate	Used to block access to dates before today.
Submit	Determines that the form will be automatically submitted when a date is chosen.
Submit=	Determines which button in the form should be selected when the date is chosen.

sessioninfo with static cache files

When using static (Fast) cache files no tags will be processed. The system will serve pages from cache. This will cause problems for sites which are using user registration where for each page the system has to show the correct user information. For this purpose Saxotech online provide the following tags which are processed when static cache files are enabled.

The processed content of all the tags mentioned here will only be added during the post processing of static cache files if the following is set in the publicus.ini:

```
[CACHE]
EnablePostProcessSessionInfo=1

without this setting these solutions will not work
```

<pbs:#sessioninfo#>

This tag is processed when using static cache files. It calls the sessioninfo function. This tag will not use any object class.

The `<pbs:#sessioninfo#>` will be processed in before dispatch thus global variables defined during the process of this tag will be available during the processing of all tags. This tag will not be processed for the following actions:
`/STATUS, /VER, /ALIVE, /EXEC`

<pbs:#@sessioninfo#>

This tag is processed when using static cache files. It calls the sessioninfo function and uses object class 999

The `<pbs:#@sessioninfo#>` will be processed in before dispatch thus global variables defined during the process of this tag will be available during the processing of all tags. This tag will not be processed for the following actions:
`/STATUS, /VER, /ALIVE, /EXEC`

<pbs:#@sessioninfoarticle#>

This tag is processed when using static cache files. It calls the sessioninfo function and uses object class 998. As the name of the tag indicates this tag is only processed for `/ARTICLE` action.

The `<pbs:#@sessioninfoarticle#>` will be processed in before dispatch thus global variables defined during the process of this tag will be available during the processing of all tags. This tag will not be processed for the following actions:
`/STATUS, /VER, /ALIVE, /EXEC`

<pbs:#sessioninfopostproc_1#>

This tag is processed when using static cache files. It calls the sessioninfo function and uses object file sessioninfopostproc.pbo with object class 1.

The `<pbs:#sessioninfopostproc_1#>` will be processed in Post processing thus global variables defined during the process of the request will be available in this tag. This also means that any global variable created in this tag is not available during the process.

<pbs:#sessioninfopostproc_2#>

This tag is processed when using static cache files. It calls the sessioninfo function and uses object file sessioninfopostproc.pbo with object class 2.

The `<pbs:#sessioninfopostproc_2#>` will be processed in Post processing thus global variables defined during the process of the request will be available in this tag. This also means that any global variable created in this tag is not available during the process.

The processed content of all the tags mentioned above will only be added during the post processing of static cache files if the following is set in the publicus.ini:

Community Site

Community Site (previously known as Publicus Light) is the Community Publishing Module, which allows your customers to maintain their own sites as a part of your website. Your customers have access to the same functions that you have.

A Community Site can use the following features:

- Stories
- Links

All other tags will pull information from the main site. A Community Site can also pull information from a main siteÕs database. This allows the site to show stories from a category or stories tagged with a keyword or a theme.

- Community site setup
- Other tags and community site
- pbs article and community site
- pbs buildmenu
    - MenuGraphicItem.pbo
    - MenuTextItem.pbo
- pbs lightid
- pbs links on community site
- pbs sectionfront on community site
- Community site templates

Community site setup

To enable Community Site, the option

```
light=1
```

must be set in your Publicus.ini in the MODULES section.

Create Users

To create a Community Site user, you must log on to the editing interface as a system administrator.

When you create the new user, you must use the **Light ID** field. This ID field is a numerical value from 001 to 999. At least one user for each light ID must be given system administrator rights. These rights apply only to the Community Site, and not the entire website.

Instructions for creating users can be found in the *SAXOTECH Online User Manual.*

Create categories

After you have created a user, the user must log on and create categories. These categories can have any names. The main category must be named FRONTPAGE. These categories are separate from the main siteÕs categories.

Alias

To give the users easier access to the Community Sites hosted by you, you should create an alias that redirects users to the main page. If you have created a Community Site customer with ID 001 and the category FRONTPAGE for the front page, the alias must point to:

`/apps/pbcs.dll/section?Category=FRONTPAGE&LID=001`

Other tags and community site

All other tags will pull information from the main site's database.

pbs article and community site

<pbs:article>

The article tag does not support object files on Community Sites.

pbs buildmenu

<pbs:buildmenu>

Used to build Community Site menus, this tag builds a sorted menu based on the value in the **Sort** field in the category register.

MenuGraphicItem.pbo

<%link%>	The link to the section front.
<%Image%>	Used to show an image based on the category. It is used like this:{{<a href="<%link%>"><img src="<%Image%>" border=0> }}The system looks for an image called **menu _<category>.gif** in the folder: **/light/<LightID>/graphics**

Example

```
<a href="<%link%>"><img src="<%Image%>" border=0><br></a><br>
```

MenuTextItem.pbo

<%link%>	The link to the section front.
<%Categoryname%>	The name of the category.

Example

```
<a href="<%link%>" class="menu" style="color: #000000;">
<%Categoryname%></a><br>
```

pbs lightid

<pbs:lightid>

This tag has no options.

pbs links on community site

<pbs:links>

The links tag supports object files on Community Sites.

pbs sectionfront on community site

<pbs:sectionfront>

Builds a section front for the Community Site. The tag follows normal behavior.

Community site templates

The templates which are used by Community Site are:

gen_ovr.pbs
gen_art.pbs

You can also create templates for each category. These templates follow default SAXOTECH Online behavior. Community Sites do not use the **fr ontpage.pbs** template.

The templates for each Community Site reside in this folder:

```
/light/<customerid>
```

Xsendmail

XSendMail is a component that handles forms on web pages. It takes information in the form and translates it into an e-mail, which is sent to the specified address.

To use XSendMail, the XSendMail.ini file must be present in the folder above your application folder.

- XSendMail.pbo
- XSendMail Form
- XSendMailHTML.pbo
- XSendMailHTMLItem.pbo
- XSendMailItem.pbo
- XSendMail setup

XSendMail.pbo

<%Copyright%>	Copyright message
<%SentMessage%>	Returns 'This form was successfully sent to #Recipient-Mail#'. Only in the receipt mail.
<%ImpMessage%>	The "Do not reply to this email message" message.
<%Items%>	Returns the values from the XSendMailItem.pbo.
<%Body%>	Returns the body text
<%Title%>	Only used for XSendMailHTML.pbo. The result of the title field from the Send Mail form.

XSendMail Form

The form for XSendMail has the following structure:

```
<form method="post" action="/apps/XSendMail.dll">
Fields go here.
Submit button.
</form>
```

Templates

The form can be put on all pages. The example template is called **XSendMail_ovr.pbs**.

Fields

You can create any fields you like, as long as they do not have the same names as any of the hidden fields.

Hidden fields

These fields are put in the form as hidden fields.

Body	This is a text field that is put in the e-mail. By default, this field has no value. To use this field, enter a value similar to the following `<textarea rows="2" cols="20" name="body">This is a message</textarea>`

File	Allows web guest to select a local file to be sent with the form.
	
	`<input type="file" name="fileupload">`
	
	When you will use file upload, you need to add this to your form code enctype="multipart/form-data"
	
	`<form enctype="multipart/form-data">`

From address	Hides the sender address. By default, the setting in the server-wide XSendMail.ini is used. This value must be a valid mail address. Enter the following to hide the sender's address: `<input type="hidden" name="from" value="server@website.com">`
From name	Hides the sender name. By default, the setting in the server-wide XSendMail.ini is used. Enter the following to hide the sender's name: `<input type="hidden" name="fromname" value="The Webserver">`
Hide recipient	Disables the output of the recipient's e-mail address in the e-mail sent to the person filling in the form. By default, the e-mail address is shown. Enter the following to suppress the recipient's address: `<input type="hidden" name="HideRecipient" value="1">`
Language	Sets the language for the e-mail and system messages. It is typically used like this: `<input type="hidden" name="lang" value="XXX">` By default, the language is Norwegian. Other languages available use the following values: • **NOR** Norwegian • **SWE** Swedish • **ENG** English • **DAN** Danish • **POL** Polish

No HTML	Used to disable HTML in the e-mail. By default, e-mail is output as HTML. To send e-mail as plain text use the following: `<input type="hidden" name="nohtml" value="1">`
No receipt	Determines whether a receipt is sent to the sender. By default, a receipt is sent if there is added a Reply-To mail adress. Enter the following if no receipt should be sent to the sender: `<input type="hidden" name="noreceipt" value="1">`
Objectclass	Sets the object class. By default, object files are not used. Objectclass is set as follows: `<input type="hidden" name="objectclass" value="23">`
Reply-To	This option is used to set the reply to mail adress. `<input type="hidden" name="REPLY-TO" value="mail@domain.com">`
Required	Determines which fields are required. The web guest will view missing fields listed on an error page. It is important to use field names that make sense to the web guest.By default no fields are required. Enter the following to set required fields: `<input type="hidden" name="REQUIRED" value="To,From,Name,E-mail">`
Return_link	This option is used to set the return link on the error page. It is used as follows: `<input type="hidden" name="Return_link" value="javascript:history.go(-1)">` By default the value is javascript:history.go(-1). You can use any URL for the return address.
Return_text	Sets the text for the return link. By default the return text is "Tilbake til skjema." This can be changed to any text string.

Redirect_URL	Sets the URL to which the web guest is directed after the form is filled in and submitted correctly. This option is required.By default, this option has no value. Enter a value similar to the following: `<input type="hidden" name="REDIRECT_URL" value="/misc/thankyou.htm">`
Subject	Sets the subject for the e-mail. By default, the field has no value. Enter the following to create a subject: `<input type="hidden" name="SUBJECT" value="Your Subject Text">`
Title	Sets the title for the form. The field is used like this: `<input type="hidden" name="TITLE" value="A test form">`
To	Sets the e-mail address to which the form is sent. This option must be set using the following tag: `<input type="hidden" name="TO" value="someone@website.com;someone2.website.com">` If you want the e-mail to be sent to multiple recipients, you can specify multiple e-mail addresses in a semicolon separated list. They are delimited by either a ";" or ",".

Validatekey	The validatekey validates against spam.
	Key verification is enabled by default so this line must be added to your form code.
	
	The option is activated using the following:
	
	`<input type="hidden" name="validatekey" value="<pbs:!validatetag>">`
	
	Note that two settings must be set in the XSendmail.ini in the GENERAL section:
	
	`KeyVerificationEnabled=1 (Default: 1)`
	`KeyVerificationLifeTime=X`
	
	where X is the number of minutes for the lifetime of the key.

Captcha	Xsendmail support Captcha. Read more here pbs Captcha `<pbs:!captcha Length=5>` In addition, the BypassValidation now bypasses all validation schemes. So BypassValidation must be turned off.
UTF-8 **From Version 7.0.0sr16 and 7.1.0sr4**	Used to inform the dll, that the input text is in UTF-8 format. This is mainly used when submitting the form using an Ajax request.

XSendMailHTML.pbo

This file is used for the HTML version of the e-mail. It has the same fields as XSendMail.pbo.

XSendMailHTMLItem.pbo

This file is used for the HTML version of the email. It has the same fields as XSendMailItem.pbo.

XSendMailItem.pbo

<%Name%>	The name of the field.
<%Value%>	The value of the field.

XSendMail setup

Xsendmail requires its own ini file called **default_XSendMail.ini**. To make a site specific version, follow the standard SAXOTECH Online logic.

GENERAL section

The XSendMail.ini GENERAL section has the following settings:

Captcha	The user's solution to the captcha. Protects forums using a captcha using the <pbs:captcha> tag. To enable this setting, enter `Captcha  = 1` The default is 0. The form submitted must contain the CaptchaID value.
BypassValidation	BypassValidation bypasses all types of validation (+ Captcha). Default is 0.
BypassVerification	When enabled, the three verification steps for XSendMail are disabled for testing purposes. To enable, enter `BypassVerification=1` The default is disabled.
NoOfEmailsToBeSent	Control the number of email sent at one time.
From	Sets the default email address from which messages are sent.
Fromname	Sets the default sender name.
KeyVerificationEnabled	Used to turn key verification on and off. The default is 1 (enabled).
KeyVerificationLifetime	The key lifetime in minutes. (0...N). The default is 30.

Mailserver	The fully qualified address is the address of the SMTP server that will send the email. You can also use the IP address instead of the address.
ReferrerVerificationEnabled	Used to turn referrer verification on and off. The default is 1.
ReferrerVerificationMethod	Used to determine which approach is used when trying to match the allowed referrers to the actual referrers. Note that all referrer matching is case sensitive. For example: `ReferrerVerificationMethod=[REGEXP*|INSTR|HEAD|EXACT]` Where: • REGEXP matches by searching for the allowed referrer as a regular expression • INSTR matches by searching for the allowed referrer in the referrer string • HEAD matches the allowed referrer to the head/start of the actual referrer • EXACT performs an exact match
ReferrerVerificationSection	The name of the section in the ini file where the allowed referrers are listed. It defaults to REFERRERS, which is the list used by the old verification routine. `ReferrerVerificationSection=[REFERRERS*|<newsection>]`
ReferrerVerificationShowDetails	Used to make XSendMail report details on why verification failed, including the actual referrer and the valid referrers loaded from the ini file. The default is 0 (disabled).
UseNewVerificationScheme	Set to 1 to enable the new verification scheme. The default is 0. The verification scheme performs the following three checks: • Checks to make sure the referrer simply matches the domain from Publicus.ini. Referrer is the Request.Referrer. • If it does not directly match the domain-setting, the same thing is checked for all values in the REFERRERS section in the XSendMail.ini. Then the systems checks whether the allowed host has been added to the ini-file.

Example:

```
[GENERAL]
Mailserver=fully.qualified.address
From=webserver@somesite.com
Fromname=The Webserver
UseNewVerificationScheme=0
KeyVerificationEnabled=1
KeyVerificationLifeTime=30
ReferrerVerificationEnabled=1
ReferrerVerificationShowDetails=0
ReferrerVerificationMethod=REGEXP
ReferrerVerificationSection=REFERRERS
```

REFERRERS section

This section is used to list the actual referrers. For example:

```
[REFERRERS]
www.mydomain.com
http://www.anotherdomain.com
```

Go.dll

This is a method for tracking clicks.

Use	`<a href="/apps/go.dll?http://www.publicus.com/">Go! Go!</a>`
Logging	The clicks are logged in the folder **/adlogs.** The log file is called ads *<isodate>_<number>.log*

guide_GuideAdItem

GuideAdItem.pbo

The GuideAdItem makes a list of the ads for a given guide.

<%Active%>	Is the item active?
<%ActiveImageCounter%>	Number of images in active items.
<%AdDescription%>	Description of the ad item.
<%AdID%>	ID of the ad item.
<%AdText%>	Text of the ad item.
<%AdTitle%>	Title of the ad item.
<%EndDate%>	Last appearance date in the guide for an ad item.
<%GuideAdShowFullSize%>	
<%GuideID%>	ID of the guide for the ad item.
<%ImageCounter%>	Number of images in an ad item.
<%StartDate%>	First appearance date in the guide for an ad item.
<%UnsavedListIndex%>	Always -1

guide_GuideMainItem

GuideMainItem.pbo

Output for each main category in the system.

<%GuideMainCat%>	The title of the main category
<%GuideMainCatLink%>	The link to the list of all the entries in the main category.
<%GuideSubCatItems%>	Outputs all the subcategories.

Deprecated

The Tags listed under this page are deprecated and have been replaced by newer functionality.

Forum Tags

Forum Tags

The Forum module is deprecated. SAXOTECH offers commenting through the Comments module. For true forum functionality you should integrate a third-party forum.

Forums give you the ability to interact with your web guests. This function allows guests to comment on stories and sections.

There are several settings in the Publicus.ini that must be added before you can use forums on your site. Please see the FORUM section.

- Message filtering
- Message form
  - ForumNewMessage.pbo
  - ForumNewMessageExtLogin.pbo
  - ForumNewMessageUnavailable.pbo
- pbs forumoptions
  - ForumPreferences.pbo
  - ForumPreferencesSaved.pbo
  - ForumPrivateEmail.pbo
  - ForumPrivateEmailMessage.pbo
  - ForumPrivateEmailSent.pbo
- pbs news_content
  - ForumMessage.pbo
  - ForumMessageAttachmentItem.pbo
  - ForumMessageThreadItem.pbo
  - ForumThemeItem.pbo
  - ForumThemes.pbo
  - ForumThemesGroupItem.pbo
- pbs news_count
- pbs news_grouplink
  - ForumGroupLink.pbo
- pbs news_groups
  - ForumGroupHeadlineItem.pbo
  - ForumGroupItem.pbo
  - ForumGroups.pbo
- pbs news_hotthreads
  - ForumHotThreadItem.pbo
  - ForumHotThreads.pbo
- pbs news_latest
  - ForumLatestMessageItem.pbo
  - ForumLatestMessages.pbo
- pbs news_messages
  - ForumMessageItem.pbo
  - ForumMessages.pbo
  - ForumMessagesMessageThreadItem.pbo
  - ForumNavBar.pbo
  - ForumNavBarItem.pbo
- pbs search
- pbs searchresult and forums
  - SearchResultForum.pbo
  - SearchResultForumItem.pbo
- pbs userverify
  - UserVerificationMail.pbo
  - UserVerify.pbo

Message filtering

The forums support message filtering. This function is used to replace profanities and other objectionable content. The system matches whole words.

The file containing the list of blocked words is called **nolist.txt** and is placed in the root folder of the website. Enter one word per line. Phrases can also be used. The file should be no longer than 50 words.

This system is not foolproof. If your users want to use these words they will find a way around it. By replacing letters with numbers and misspelling words is possible bypass the system like this: `r3allybadw0rd`.

Message form

This form is used for all message creation and editing.

ForumNewMessage.pbo

<%PostMessageLink%>	The link to the program that processes the submission.<form method="POST" action="<%PostMessageLink%>">
<%Attachments%>	Lists file attachments. Please see ForumMessageAttachmentItem.pbo.

<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same is
	`*<%Attachments%>*`
	minus
	`*<%Images%>*`

<%Author%>	The name of the logged on user. This is usually the nickname, but if there is no nickname it is the user name.
<%AvatarLink%>	An image link to the logged on user's avatar.
<%Body%>	The body text of the current message.
<%CanPost%>	This field has a value of 1 if the user has a user level above 1.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The group name for this message
<%Edit%>	This field has a value of 1 if editing an existing message.
<%EditMessageLink%>	The link to the page for editing the current message.
<%Email%>	The logged on user's email address.
<%ExtEmailAdr%>	When extended login is used, this is the user's email address.
<%ExtFirstName%>	When extended login is used, this is the user's first name.
<%ExtSurName%>	When extended login is used, this is the user's surname.
<%FirstName%>	The logged on user's first name.
<%FromUrl%>	The referring URL.
<%GroupID%>	The ID of the group in which this message will be created.
<%GroupLink%>	A link to the list of messages for a link.

<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for for the theme in which the current message exists. This is typically used to present a drop-down list of groups that users can use to navigate to a new group.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. The **Hotthreshold** para meter on the pbs news_content tag controls the number of replies required before a message is considered hot.
<%Images%>	(Threaded Forums only) The number of image attachments associated with this message.
<%InReplyTo%>	The ID of the message to which this post responds.
<%IsGrpLocked%>	This has a value of 1 if the group in which this message belongs is locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsModerator%>	This field has a value of 1 if the logged on user is a moderator.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 when the current message has been created since the current user's last visit date/time.
<%ISOCreated%>	The same as <%Created%> except in ISO format.
<%ISOLastMessageDate%>	The same as <%LastMessageDate%> except in ISO format
<%ISOPublished%>	The same as <%Published%> except in ISO format.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%LastName%>	The logged on user's last name.
<%Link%>	The link to the full text message.
<%MessageID%>	A value the represents the current message as a number.
<%Name%>	The user name of the logged on user.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%NickName%>	The logged on user's nickname.
<%PostMessageLink%>	A link that will submit the message to the system.
<%PreviewBody%>	A preview of the body with all the graphics resolved.
<%PreviewMessageLink%>	A link that will preview the message.
<%Published%>	For moderated forums, this is the date/time the moderator published the message. For non-moderated forums this is the date/time the message was created.
<%QuoteBody%>	The quoted body of the original message.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowAllLink%>	A link to a list of messages for this messages group that is not restricted by **MaxDays.**
<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The logged on user's signature.
<%Theme%>	The theme name for this message.

<%ThemeDescription%>	The theme description for this message.
<%ThemeId%>	The ID of the theme in which this message will be created.
<%ThemeLink%>	A link to the list of groups for the theme.
<%ThreadCount%>	The number of replies to this message.
<%Title%>	If this is a reply, this field contains the title of the original posting with Re: appended to the beginning.
<%Today%>	This field has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%UserLevel%>	The user level of the logged on user.
<%ValidCaptcha%>	When a forum is protected using a captcha tag, if the user submits an incorrect solution the dll returns the preview page instead. <%ValidCaptcha%> is only available when the solution was incorrect. It has a value of 0.

New Message Form Example

```
<form method="POST" action="<%PostMessageLink%>" name="commentform"
enctype="multipart/form-data">

<input type="hidden" name="InReplyTo" value="<%InReplyTo%>">

Name:<br>
<input type="text" name="Author" maxlength=50 size=40><br><br>

Email:<br>
<input type="text" name="Email" maxlength=255 size=40 value="<%Email%>"><br><br>

Title:<br>
<input type="text" name="Title" maxlength=255 size=40 value="<%Title%>"><br><br>

Body:<br>
<textarea name="Body" rows="10" cols="30"></textarea><br /><br />

<%ValidCaptcha%>
<pbs:captcha Length="4"><br>

<input type="Submit" value="Submit">

</form>
```

Output form fields

The following form fields are sent to the application to complete the action following a form submission.

Attachments	Any HTTP file input type `<input type="file" ...>` will be interpreted for attachments.
Author	The author name that is used in non Extended Forums.
Body	Body text of the new message.

DeleteAttachments	If this item has a checked value, then any current attachments will be removed from the message.
Edit	If this value is set to 1, an existing record is to be updated.
Email	The e-mail address that is used in non Extended Forums.
EmailReplies	If this value is set to 1, the user is to be notified when someone has replied to the message.
Extra_XXX	Name and value details can be stored against a message. These details will be displayed in the editing interface when editing the message. An example:<input type="hidden" name="Extra_postcode" value="GL7 3QU">
GroupRedirect	If this value is set to 1, all the messages for the current group will be displayed after the post completes.
InReplyTo	The message ID against which a reply is set.
MessageID	The ID of the message being edited. See Message form.
ModeratorMsg	If this value is set to 1, this message will be marked as a moderator message.
MsgType	If this value is set to 1, this message will be marked as a sticky message.
Signature	If this value is set to 1, the user's signature will be stored with this message. Extended Forum option.
Title	The message title.

ForumNewMessageExtLogin.pbo

This file is rendered in preference to ForumNewMessage.pbo when extended login is enabled. ForumNewMessageExtLogin.pbo has the same variables as ForumNewMessage.pbo

ForumNewMessageUnavailable.pbo

This file is rendered if there is no write access to the database. ForumNewMessageUnavailable.pbo has the same variables as ForumNewMessage.pbo

pbs forumoptions

<pbs:forumoptions>

The user preference function is used to set the preferences that apply to the Forum Module in SAXOTECH Online.

Template

Gen_forumoptions.pbs.

ForumPreferences.pbo

This object file brings up the current information, in addition to saving the data input by the user. The output fields are listed with their input fields.

Input fields

Avatar	Used to attach a picture to the user information. This picture can then be shown with every post the user makes. `<input type="file" name="Avatar"> <img src="<%avatarlink%>&maxw=100&NoCache=1">` Delete `<input type="checkbox" name="DeleteAvatar" value="1">`
First name	Captures the first name for the user. `<input type="text" name="FirstName" value="<%FirstName%>">`
Form setup	The form must be set up so the posted information will be saved. `<form method="POST" Action="<%pbcs.dll%>/forumoptions?Category=<%Category%>&SavePreferences">`
Last name	Captures the user's last name. `<input type="text" name="LastName" value="<%LastName%>">`
Nickname	Captures a nickname. The nickname is displayed instead of the user's real name in the forum listing. `<input type="text" name="NickName" value="<%NickName%>">` The NickName entered should not consist of blank spaces.

Signature	Captures the signature. It will be attached to every posting. `<textarea name="Pref_Signature" rows=6 cols=45 > <%Signature%> </textarea>`

Output fields

<%Author%>	The currently logged in user's nickname. If there is no nickname, the first name followed by the last name.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%EMail%>	The currently logged in user's email address.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPreferencesSaved.pbo

<%\Author%>	The currently logged in user's nickname. If there is no nickname, then it is the first name followed by the last name.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the objects directory.
<%\EMail%>	The currently logged in user's email address.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPrivateEmail.pbo

Used to allows users to e-mail each other when the user's email addresses are not shown on the forum page.

This is used in conjunction withForumPrivateEmailMessage.pbo and ForumPrivateEmailSent.pbo

URL

The URL for the form is generated in the message listing and should look something like this:

```
/apps/pbcs.dll/forumoptions?Category=forum
*&Privateemail=<%PostedBy%>*
```

Input fields

Form	The form must be set up like this: `<form method="POST" Action="<%pbcs.dll%>/forumoptions?Category=<%Category%>&SendPrivateEmail">`
To Address	The to address must be set up in a hidden field like this: `<input type="hidden" name="EMailTo" value="<%EMailTo%>">`
From Name	The sender's name. `<input type="hidden" name="FromName" value="<%Name%>">`
From Address	The sender's address. `<input type="hidden" name="FromAddress" value="<%\EMail%>">`
Email subject	Captures the email subject. `<input type="text" name="Subject" maxlength=255 size=60 class=forum>`
Message body	The message body. `<textarea name="MessageBody" rows=20 cols=45 class=forum></textarea>`

Output fields

<%\Author%>	The name of the user that created the message.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The currently logged in user's e-mail address.
<%EmailTo%>	The email address of the user that created the message.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPrivateEmailMessage.pbo

Used to allows users to e-mail each other when the user's email addresses are not shown on the forum page.

This is used in conjunction with ForumPrivateEmail.pbo and ForumPrivateEmailSent.pbo

<%\Author%>	The currently logged in user's nickname. If there is no nickname, then it is the first name followed by the last name.
<%AvatarLink%>	The currently logged in user's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The currently logged in user's email address.
<%FirstName%>	The currently logged in user's first name.
<%IsModerator%>	This is set to 1 if the currently logged in user is a moderator.
<%LastName%>	The currently logged in user's last name.
<%Name%>	The currently logged in user's first name followed by the last name.
<%NickName%>	The currently logged in user's nickname.

ForumPrivateEmailSent.pbo

Used to allows users to email each other when the user's email addresses are not shown on the forum page.

This is used in conjunction with ForumPrivateEmail.pbo and ForumPrivateEmailMessage.pbo.

<%\Author%>	The recipient name.
<%AvatarLink%>	The recipient's avatar image link.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%\EMail%>	The email address of the recipient.
<%FirstName%>	The recipient's first name.
<%FromAddress%>	The email address of the currently logged in user.
<%FromName%>	The name of the currently logged in user.
<%IsModerator%>	This is set to 1 if the recipient is a moderator.
<%LastName%>	The recipient's last name.

<%MailSent%>	Has a value of 1 if the mail is sent successfully.
<%MessageBody%>	The message body of the private mail that was sent.
<%Name%>	The recipient's first name followed by the last name.
<%NickName%>	The recipient's nickname.
<%Subject%>	The message subject of the private mail that was sent.

pbs news_content

<pbs:news_content>

Used to show the list of themes on your site and to view messages.

Template

The default template for the forums is **gen_news.pbs**. You can also use this tag on other templates.

URL

The main URL for forums is **/apps/pbcs.dll/news_themes?Category=FORUM**.
The **Category** is optional, but should be specified so you can view statistics on forum traffic.

Options

All options are listed here, even if they are used on other pages than the theme listing.

Ascthreadsort	Makes the message threads sort in ascending order on the messages page. The default is to sort the message threads in descending order.
Conditional	Enables the listing of theme and article forums. Without this option, only group forums are shown.
Expandthreads	Enables the expansion of threads in the messages overview so you can see all stories in the thread you are viewing. This option cannot be used with a protected forum.
Hotthreshold	Sets the threshold for when a message thread is hot. The default threshold is 10 replies in 24 hours. Enter `hotthreshold=25` if the message must have 25 replies in 24 hours before it is considered hot.

Maxdays	Sets how many days back the tag should look for forum postings. By default, this option is not used.To set the maximum days the tag should look back for forum postings, enter
	`maxdays=X`
	where
	`X`
	is the number of days.

ShowHeaders	Enables message headers in the group listing. This option can only be used if you have very few messages. Showing headers on the group page can make your site fail. This option is turned off by default. Enter `showheaders=1` to enable this option.

Threadcount	Sets the maximum number of threads that should be shown on the overview page. You can use previous and next links to navigate to the other threads. By default, all threads are shown on one page. Enter
	
	`threadcount=X`
	
	for
	
	`X`
	
	threads to be shown per page. You must use the functions for previous and next links. Otherwise, web guests cannot navigate to all message threads.

ForumMessage.pbo

This function displays the actual message. The link to this page is automatically built by the system with the pbs news_content tag.

Template

The default template for forums is **gen_news.pbs**. The template is accessed through this URL:

```
/apps/pbcs.dll/news_message?ThemeID=1&GroupID=7&InReplyTo=16
```

Theme ID, Group ID and InReplyTo must be specified.

This is used in conjunction with ForumMessageAttachmentItem.pbo and ForumMessageThreadItem.pbo.

<%ArtLink%>	A link to the originating story if the forum is an Article Forum.
<%Attachments%>	Lists file attachments. Please see ForumMessageAttachmentItem.pbo
<%\Author%>	The name of the person who wrote the message. The user's nickname is shown if the user has registered a nickname. Otherwise, the name with which the user registered is shown.
<%AvatarLink%>	The link to the image associated with the user.
<%Body%>	The message body.
<%Created%>	The date/time the message was written.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.

<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The name/description of the group in which the message is posted.
<%EditMessageLink%>	The link to the page for editing the message. This feature is only available to the original poster.
<%\Email%>	The email address of the author.
<%GroupLink%>	The link back to the group listing.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for the theme in which the current message exists. This is typically used to present a drop-down list of groups that the user can use to navigate to a new group.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. The **Hotthreshold** para meter on the <pbs:news_content> tag controls the number of replies required before a message is considered hot.
<%Images%>	Lists image attachments.
<%IsGrpLocked%>	This field has a value of 1 when the group has been locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 when the current message has been created since the current user's last visit date/time.
<%IsoCreated%>	The date/time the message was written, in ISO format
<%ISOLastMessageDate%>	The same as `<%LastMessageDate%>` except in ISO format.
<%IsoPublished%>	The date/time the message was published, in ISO format. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%ISOUserRegDate%>	The date the user was added to the user database.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%LastPoster%>	The nickname of the user that last posted to this thread.
<%Link%>	The link to the full text message.
<%MessageID%>	The ID for the message.
<%MessageThreads%>	Lists replies to this message in a threaded style forum. By including this variable, you define your forum as a threaded forum.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%PostedBy%>	The numeric User ID.
<%PostedByHost%>	The IP address of the user who posed the message.

<%Published%>	The date/time the message was published. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%Replies%>	Lists replies to this message in a non-threaded style forum. By including this variable you define your forum as a non-threaded forum.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The user's signature.
<%Theme%>	The name of the theme in which the message is posted.
<%ThemeDescription%>	The theme description for this message.
<%ThemeLink%>	A link to the list of groups.
<%ThreadCount%>	The number of replies to this message.
<%\Title%>	The title of the message.
<%Today%>	This has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%UserLevel%>	This field contains the user level.
<%UserPostCount%>	The total number of posts by the user.
<%LoggedInUserVerified%>	This field has a value of 1 if the currently logged on user is verified.
<%LoggedInUserLevel%>	The user level of the currently logged on user.

ForumMessageAttachmentItem.pbo

This file is used to list all attachments for a given forum message. It is used in conjunction with ForumMessage.pbo and ForumMessageThreadItem.pbo.

<%Counter%>	The counter for the attachments.
<%DownloadLink%>	The link that will download the attachment.
<%Filename%>	The file name of the attachment.
<%ImgLink%>	The link that will display the attachment, if it is an image.
<%IsImage%>	This field has a value of 1 if the attachment is an image.
<%IsLastItem%>	This field has a value of 1 if the attachment is the last in the list.

ForumMessageThreadItem.pbo

This file is used to list all messages in the thread. It is used in conjunction with ForumMessageAttachmentItem.pbo and ForumMessageThreadItem.pbo.

It is also used on the group overview page. Please see ForumMessagesMessageThreadItem.pbo. The field

```
<%MessageThreads%>
```

must be present in the ForumMessages.pbo for this field to be used.

<%Attachments%>	Lists file attachments. Please see ForumMessageAttachmentItem.pbo.

<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same as <%Attachments%> minus <%Images%>.
<%\Author%>	The name of the person who wrote the message.
<%Body%>	The body text of the current message.
<%Category%>	The template category that is used to resolve the correct pbo files in the object's directory.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%EditMessageLink%>	The link to the page for editing the current message.
<%\Email%>	The email address of the message author.
<%GroupLink%>	A link to the list of messages for a link.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. The **Hotthreshold** para meter on the <pbs:news_content> tag controls the number of replies required before a message is considered hot.
<%Images%>	(Threaded Forums only) The number of image attachments associated with this message.
<%Indent%>	(Threaded Forums only) This field is used to place message details at the appropriate position on the screen, depending on the level of threading.
<%IsGrpLocked%>	This has a value of 1 if the group in which this message belongs is locked.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 if the current message has been created since the current user's last visit date/time.
<%ISOCreated%>	The same as <%Created% except in ISO format.
<%ISOLastMessageDate%>	The same as <%LastMessageDate%> except in ISO format.
<%ISOPublished%>	The same <%Published%> except in ISO format.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%Link%>	The link to the full text message.
<%MessageID%>	A value the represents the current message as a number.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%Published%>	For moderated forums, this is the date/time the moderator published the message. For non-moderated forums, this is the date/time the message was created.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowRepliesLink%>	The same as <%Link%>
<%ThemeLink%>	A link to the list of groups for the theme.
<%ThreadCount%>	The number of replies to this message.

<%\Title%>	The title of the message.
<%Today%>	This has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%TopThreadMessageID%>	The message ID of the original posting.
<%TreeLines%>	(Threaded Forums only) This lists tree-line characters appropriate to the level of indent for this thread item.

ForumThemeItem.pbo

Used to build each item in the list.

<%Counter%>	The counter for the theme items.
<%Description%>	The description of the theme.
<%GroupItems%>	Outputs the group items if `showgroups=1` is on the URL.
<%GroupsListLink%>	The link to the theme listing.
<%HasNew%>	This field is set to 1 if there are messages or replies created after the reader last view this page. This option uses a cookie.
<%IsLastItem%>	This field contains the value 1 if it is the last theme item listed.
<%ISOLastMessageDate%>	The last message date in ISO format.
<%LastMessageDate%>	The most recent date on which a message was posted in the theme.
<%LogoLink%>	The link for the logo, if the theme has a logo
<%MessageCount%>	The number of messages in the theme, plus all the replies.
<%Name%>	The name of the theme.
<%Theme%>	The ID for the theme.
<%ThemeLogoLink%>	This is the same as `<%LogoLink%>`.
<%Today%>	This field has the value 1 if the theme has a new message today.
<%TopMessageCount%>	The number of messages in the theme.

ForumThemes.pbo

Used to build the main theme listing.

<%ISOUserLastVisitedDate%>	Outputs the date of the last visit for this reader.
<%ThemeItems%>	Outputs the themes.

ForumThemesGroupItem.pbo

This file is used to list groups with each theme. Theme groups can be listed by entering

```
ShowGroups=1
```

on the URL, and by defining ForumThemesGroupItem.pbo.

ForumThemesGroupItem.pbo exposes the same variables as ForumGroupItem.pbo.

pbs news_count

<pbs:news_count>

This tag is used to show the number of messages in the forums. The tag will show the total number of messages on the level on which the tag is used. If the tag is used on the main overview page, it will show the total number of messages. If it is used on the group page, it will show the total number of messages for the group.

Template

This tag can be used on all templates.

Options

Group	Enables you to specify the group for which the tag should show the total number of messages. The theme must also be specified if this option is used. By default, all groups are counted.You can use pbs queryparam to set this value, or you can hardcode it. The group ID is used.
Theme	Enables you to specify the theme for which the tag should show the total number of messages. By default, all themes are counted.You can use pbs queryparam to set this value, or you can hardcode it. The theme ID is used.

pbs news_grouplink

<pbs:news_grouplink>

This tag is used to link to a forum from a story.

Options

Articleforum	Makes the system link to the article forum for the story.
Autoforum	Makes the system link to the correct forum type based on the forum setting in the story. You should use this option if you want to link from the story to the forum.
Conditional	Used with Themeforum, Articleforum and Groupforum to make these options only output a link if forum is enabled in the story.
Groupforum	Makes the system link to the group forum for the story.
Themeforum	Makes the system link to the theme forum for the story.

ForumGroupLink.pbo

<%Link%>	The link to the forum.

pbs news_groups

<pbs:news_groups>

The group listing lists all groups in a theme. The link to this page is automatically built by the system by the pbs news_content tag on the theme page.

This tag can be used on all pages. The theme number is used and must be specified when you use this tag. This tag can also be used to list a single newsgroup.

Template

The default template for forums is called **gen_news.pbs**. It is accessed through this URL:

```
/apps/pbcs.dll/news_groups?Category=FORUM&ThemeID=2
```

Theme ID must be specified. Category is optional, but should be specified so you can view statistics for the forum.

Options

Theme	Specifies the theme for which you want to list groups.

This tag also uses the same options as pbsnews_content.

ForumGroupHeadlineItem.pbo

Used to show the headlines belonging to a group directly in the group listing. If you have a lot of messages, this file will make your forum pages very slow.

<%Attachments%>	A value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same is <%Attachments%> minus <%Images%>
<%\Author%>	The name of the person who wrote the message.
<%Counter%>	This is an index value for the headline item. The count starts with a value of 1.
<%\Email%>	The email address of the message author.
<%HasNew%>	This field is set to 1 if the message was created after the reader viewed the page. This option uses a cookie.
<%Images%>	The number of image attachments associated with this message.
<%IsLastItem%>	This field contains the value 1 if it is the last field listed.
<%LastThreadDate%>	The publication date for the newest message in the thread.
<%Link%>	The link to the message.
<%ThreadCount%>	The number of messages in the thread. This field cannot be used if the option Expandthreads is used in the tag.
<%\Title%>	The title of the message.

ForumGroupItem.pbo

Used to build the item list.

<%AllMessageListLink%>	A link to the story listing that shows stories according to the FORUM section setting in the Publicus.ini.
<%Counter%>	The counter for the groups.
<%Description%>	The description of the group.
<%GroupLogoLink%>	The link for the logo, if the group has a logo.
<%HasNew%>	This field is set to 1 if there are messages or replies created after the reader last viewed the page. This option uses a cookie.

<%Headlines%>	This field can be used to show the threads in a group. This option requires that the **ShowHeaders** option is enabled. This option should not be used.
<%ISOLastMessageDate%>	The same as **LastMessageDate**, but in ISO format.
<%ISOUserLastVisitedDate%>	The date this reader last visited the page.
<%IsGrpLocked%>	This field is set to 1 if the group has been locked.
<%IsLastItem%>	This field contains the value 1 if it is the last field listed.
<%LastMessageDate%>	The date of the last posted message.
<%LogoLink%>	The link for the logo, if the group has a logo.
<%MessageCount%>	The number of messages posted.
<%MessageListLink%>	A link to the story listing.
<%Moderator%>	The name of the moderator
<%Name%>	The name of the group.
<%NewMessageLink%>	A link to the new message page.
<%TopMessageCount%>	The number of messages in the group.

ForumGroups.pbo

Shows the groups belonging to a theme.

<%GroupItems%>	Lists all the group items.
<%ISOUserLastVisitedDate%>	Contains the date of the last visit for this reader.
<%ShowAllLink%>	A link showing all messages, regardless if they are older than the number of days set in the ini file for deactivation of threads. It is used like this: `<a href="<%showalllink%>">Show all messages including inactive</a>` This option can only be used if the option ShowHeaders is used in the tag.
<%Theme%>	The name of the theme being viewed.
<%ThemeDescription%>	The description of the parent theme.
<%ThemeID%>	The ID of the parent theme.
<%ThemeListLink%>	The link to the main listing of themes.
<%ThemeLogoLink%>	A link to the themes logo image, if it has one.

pbs news_hotthreads

<pbs:news_hotthreads>

This tag is used to list messages that have the most activity.

Options

Category	Sets the category to be used when showing the message. This is done to show the forum messages in a category other than the category on which the messages is listed. By default this is not set. Enter `category=categoryid` to show the message using the template corresponding to the category ID.
Count	Controls how many messages the tag will show. The default is 10.The tag can show from 1 to 99 messages.
Group	Limits the groups from which the tag should show hot messages. The default is to use all groups. Enter `group=groupnumber` The group number can be found by looking at the links in the group listing for a theme.

Maxdays	Sets how many days back the tag should look for forum postings. By default, this option is not used.To set the maximum days the tag should look back for forum postings, enter
	
	`maxdays=X`
	
	where
	
	`X`
	
	is the number of days.

ObjectClass	Sets the object class. By default this is not used. Enter
	
	`objectclass=X`
	
	to use class
	
	`X`
	
	is used. Legal values are 1 through 99.

Theme	Limits the themes from which the tag should show hot messages. The default is to show all themes when viewing any page that is not showing a theme or a category. When viewing a theme or a category, the tag will only list new messages in the theme or category. Enter
	
	`theme=themenumber`
	
	The theme number can be found by looking at the links in the main forum listing. If
	
	`theme=0`
	
	is used, all messages will be shown on all pages.

ForumHotThreadItem.pbo

<%Attachments%>	A value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. This is the same as <%Attachments%> minus <%Images%>.
<%\Author%>	The author of the message.
<%Counter%>	The message counter for the list. This field is increased by one for each message.
<%Created%>	The date the message was created.
<%\Email%>	The e-mail address of the message author.
<%Group%>	The group under which the message is posted.
<%GroupLink%>	The link to the group under which the message is posted.
<%Images%>	The number of image attachments associated with this message.
<%IsLastItem%>	This field lists all entries.
<%Link%>	The link to the message.
<%MessageId%>	A value that represents the current message as a number.
<%MessageText%>	The message text.
<%Replies%>	The number of replies to the message.
<%Theme%>	This field contains the theme under which the message is posted.
<%ThemeLink%>	This field contains the link to the theme under which the message is posted.

ForumHotThreads.pbo

<%items%>	This field lists all entries.

pbs news_latest

<pbs:news_latest>

This function lists the newest messages in the forums.

Best practice:

Is to add this tag in a macro, because then the same cache file will be reused, instead of it always generates a new one, example:
template: <pbs:!macro name="getnewslatest" CacheTime="15">
\macro\getnewslatest.pbo: <pbs:news_latest count=5 showref="0">

Options

Category	Sets the category to be used when showing the message. This is done to show the forum messages in a category other than the category on which the messages are listed. By default the category is not set. Enter `category=categoryid` to show the message with the template associated with the category ID.
Count	Controls how many messages the tag will show. The default is 10. Enter `count=X` to show X messages. Legal values are 1 to 99.
Group	Limits the groups from which the tag should show newest messages. The default is to show newest messages from all groups. To use this option, set `group=groupnumber` The Group number can be found by looking at the links in the group listing for a theme.
ObjectClass	This option sets the object class. By default the object class is not set. Enter `objectclass=X` to use class X.Legal values are 1 through 99.
Showref	Controls how many characters of the message should be shown by the tag. The default is 0.The tag can be set from 0 to 999.

Theme	Used to list the newest messages for the specified theme only. By default, this setting is not used. Enter
	
	`theme=themenumber`
	
	to use this option. The theme number can be found by looking at the links in the main forum listing. Example:
	
	`<pbs:news_latest Category=phpforums_rss Count=3 showref=120 theme=1201>`
	
	To list the newest messages for the entire forum, do not use the theme parameter. Example:
	
	`<pbs:news_latest Category=phpforums_rss Count=3 showref=120>`

ForumLatestMessageItem.pbo

<%Attachments%>	Contains a value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non image attachments associated with this message. The same as `<%Attachments%>` minus `<%Images%>`
<%Author%>	The author of the message.
<%Counter%>	The message counter for the list. This field is increased by one for each message.
<%Created%>	The date the message was created.
<%Description%>	The description of the group.
<%Email%>	The email address of the message author.
<%Group%>	The group under which the message is posted.
<%GroupLink%>	The link to the group under which the message is posted.
<%Images%>	The number of image attachments associated with this message.
<%IsLastItem%>	This field contains the value 1 if it is the last theme item listed.
<%IsNew%>	This field is set to 1 if the message was created after the reader last viewed the page.This option uses a cookie.
<%Link%>	The link to the message.
<%LinkTop%>	The link to the top message.
<%MessageId%>	A value that represents the current message as a number.
<%MessageText%>	The message text. The amount of text is limited by the **showref** optio n in the tag.
<%Theme%>	The theme under which the message is posted.
<%ThemeDescription%>	The description of the theme.
<%ThemeLink%>	The link to the theme under which the message is posted.

<%Title%>	The title of the message.
<%Updated%>	The date the thread was last updated.

ForumLatestMessages.pbo

<%items%>	This field lists all entries.

pbs news_messages

<pbs:news_messages>

The message listing lists all messages in the group. The link to this page is automatically built by the the system by the pbs news_content tag on the groups page.

The tag is is used like this:

```
<pbs:news_messages articleforum objectclass=2 conditional>
```

You can also list forum messages on a story page if the forum is attached to a category or a story.
`<pbs:news_messages>` should be used uncached

```
<pbs:!news_messages>
```

when it is used on article templates.

Template

The default template for forums is **gen_news.pbs**. It is accessed through this URL:

```
/apps/pbcs.dll/news_messages?ThemeID=2&GroupID=3&Category=FORUM
```

Theme ID and group ID must be specified. Category is optional.

Options

Articleforum	Used to enable the Article Forum. It must be present when you use the tag on a story template.
photogalleryforum	Used to enable the forum in photo galleries. It must be present when you use the tag on a photogallery template.
cceforum	Used to enable the forum in CCE. It must be present when you use the tag on a CCE template.
Autoforum	Lists all the messages in the forum type based on the forum setting in the story.
Class	Sets the object class.
Conditional	Makes the forum only appear on stories where forum is enabled.
Groupforum	Lists all the messages in the group forum for the story.
Themeforum	Lists all the messages in the theme forum for the story.
Threadcount	Sets the maximum number of threads to be shown. You can use previous and next links to get to the other threads. By default, all threads are shown on one page.

ForumMessageItem.pbo

This object file is used for every top-level message.

Please note only the Extended Forum can use all the fields. The Extended Forum is set up in the Publicus.ini file. For more information, please see VerifyUsers. If Extended Forum is **not** enabled, the following fields will not be available:

```
<%AvatarLink%>
<%ISOUserRegDate%>
<%PostedBy%>
<%PostedByHost%>
<%Signature%>
<%UserLevel%>
<%UserPostCount%>
<%UserRegDate%>
```

<%Attachments%>	This field contains a value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non-image attachments associated with this message. This is the same is <%Attachments%> minus <%Images%>.
<%Author%>	The name of the person that wrote the message.
<%AuthorName%> **From version 6.5.0sr29**	The first- and lastname of the person that wrote the message.
<%AvatarLink%>	A link to the author's avatar image. This is blank if the author does not have an avatar image.
<%Body%>	Outputs the body of the message.
<%Created%>	The date/time the message was written.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The group name for this message.
<%EditMessageLink%>	The link to the page for editing the current message.
<%Email%>	The email address of the author.
<%GroupLink%>	Link to the list of messages for a link.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for the theme in which the current message exists. This is typically used to present a drop-down list of groups that the web guest can use to navigate to a new group.
<%HotThread%>	This field has the value 1 if the thread is hot.
<%Images%>	The number of image attachments associated with this message.
<%IsGrpLocked%>	This has a value of 1 if the group in which this message belongs is locked.
<%IsModerated%>	The group in which this message exists is moderated.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsoCreated%>	The date/time the message was written, in ISO format.
<%IsoPublished%>	The date/time the message was published, in ISO format. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%ISOUserRegDate%>	The same as <%UserRegDate%> but in ISO format.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.

<%LastMessageDate%>	Outputs the date of the newest reply to the message.
<%Link%>	The link to the full text of the message.
<%MessageID%>	A value that represents the current message as a number.
<%MessageThreads%>	The listing of replies to the message. This field cannot be used with the Protected Forum option.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%NickName%> **From version 6.5.0sr29**	The nickname of the person that wrote the message, if available.
<%NoVoteCount%>	Number of users who have voted no.
<%NoVoteLink%>	No vote link. The user must be logged in before voting
<%PostedBy%>	The numeric user ID.
<%PostedByHost%>	The IP address of the user who posed the message.
<%Published%>	The date/time the message was published. This value is used when the forum is moderated. If moderation is not enabled, creation date/time and published date/time will be the same.
<%ShowAllLink%>	A link to a list of messages for this message group that is not restricted by **MaxDays**.
<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The signature text of the message author.
<%Theme%>	The theme name for this message.
<%ThemeDescription%>	The theme description for this message.
<%ThemeLink%>	A link to the list of groups.
<%Title%>	The title of the message.
<%ToggleLockLink%>	A link that will cause the messages locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.
<%UserId%>	The id of the user.
<%UserLevel%>	The user level of the message author.
<%UserPostCount%>	The total number of posts the author has made in the system.
<%UserRanking%>	The ranking given by the specific user on an object.
<%UserRegDate%>	The date the user first registered in the system.
<%YesVoteCount%>	Number of users who have voted yes.
<%YesVoteLink%>	Yes vote link. The user must be logged in before voting.

ForumMessages.pbo

This file builds the framework for the message list.

<%ArtLink%>	A link back to the story, if the forum is an Article Forum.
<%Description%>	The description of the group.
<%GroupLogoLink%>	The same as <%LogoLink%>.
<%GroupsListLink%>	The same as <%ThemeLink%>.
<%IsGrpLocked%>	This field is set to 1 if the group is locked.
<%IsModerated%>	This field is set to 1 if the group is moderated.

<%LoggedInUserLevel%>	The user level of the user currently logged on to the forum.
<%LoggedInUserVerified%>	This field is set to 1 if the user currently logged on is verified.
<%LogoLink%>	The link for the logo, if the group has a logo.
<%Messageitems%>	This parameter lists all the top level messages in the group.
<%NavBar%>	Lists a page navigation area. This functionality provides a more powerful alternative to using NextLink, PageNum, NumPages and PrevLink.
<%NewMessageLink%>	The link to the new message page.
<%NextLink%>	The link to the next page. This option is used when you use **Threadcount** to limit the number of threads on one page.
<%NumPages%>	The number of pages that are required to display all the messages. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PageNum%>	When messages are displayed across several pages, this field contains a numeric value for the page index currently displayed. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PrevLink%>	The link to the previous page. This option is used when you use **Threadcount** to limit the number of threads on one page.
<%ShowAllLink%>	A link showing all messages, including those older than the cutoff date in the Publicus.ini. The link is used like this: `<a href="<%showalllink%>">Show all messages including inactive</a>`
<%Subject%>	The description of the group.
<%Theme%>	The name of the theme.
<%ThemeDescription%>	The description of the theme.
<%ThemeLink%>	The link back to the group listing.
<%ThemeLogoLink%>	The link for the theme logo, if the theme has a logo.

ForumMessagesMessageThreadItem.pbo

This object file is used to list the replies to a top level message. If the file does not exist, ForumMessageThreadItem.pbo will be used.

Please note only the Extended Forum can use all the fields. The Extended Forum is set up in the Publicus.ini file. For more information, please see VerifyUsers. If the Extended Forum is not enabled, the following fields will **not** be available:

```
<%AvatarLink%>
<%IsAdminUser%>
<%IsModerator%>
<%IsNormalUser%>
<%ISOUserRegDate%>
<%IsROnlyUser%>
<%LastPoster%>
<%PostedBy%>
<%PostedByHost%>
<%Signature%>
<%UserID%>
<%UserLevel%>
<%UserPostCount%>
<%UserRegDate%>
```

<%Attachments%>	A value for the number of file attachments that are associated with this post.
<%AttachmentsNotImage%>	The number of non-image attachments associated with this message. This is the same is `<%Attachments%>` minus `<%Images%>`.
<%\Author%>	The name of the person who wrote the message.
<%AvatarLink%>	A link to the author's avatar image. This is blank if the author does not have an avatar image.
<%Body%>	The body of the message.
<%Created%>	The date/time the message was written.
<%DeleteMessageLink%>	A link to an action that will cause the top message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%DeleteReplyLink%>	A link to an action that will cause the current message to be deleted with all its sub messages. Deleting messages actually only marks those messages as unpublished.
<%Description%>	The group description of this message.
<%EditMessageLink%>	The link to the page for editing the current message.
<%\Email%>	The email address of the author.
<%GroupLink%>	A link to the list of messages for a group.
<%GroupOptions%>	This field contains an HTML option fragment that contains a list of group names and IDs for for the theme in which the current message exists. This is typically used to present a drop-down list of groups, that the user can use to navigate to a new group.
<%HasNew%>	This has a value of 1 when the current message has replies that have been added since the current user's last visit date/time.
<%HotThread%>	This field has the value 1 if the thread is hot. Note the **Hotthreshold** parameter on the pbs news_content tag controls the number of replies required before a message is considered hot.
<%Images%>	The number of image attachments associated with this message.
<%Indent%>	Controls the story indent.
<%IsAdminUser%>	This value is 1 if the author of the message has a user level of 9.
<%IsGrpLocked%>	This field has a value of 1 when the group has been locked.
<%IsModerated%>	The group in which this message exists is moderated.

<%IsModerator%>	This value is 1 author of the message has a user level of 8.
<%IsMsgLocked%>	This has a value of 1 if the message is locked.
<%IsNew%>	This has a value of 1 when the current message has been created since the current user's last visit date/time.
<%IsNormalUser%>	This value is 1 if the author of the message has a user level in the range of 5 to 7.
<%ISOCreated%>	Same as <%Created%> except in ISO format.
<%ISOLastMessageDate%>	The same as <%LastMessageDate%> except in ISO format.
<%ISOPublished%>	The same as <%Published%> except in ISO format.
<%ISOUserRegDate%>	The same as <%UserRegDate%> but in ISO format.
<%IsROnlyUser%>	This value is 1 if the author of the message has a user level of 1.
<%IsSticky%>	This has a value of 1 if the message is a sticky message. A sticky message will appear above all normal messages.
<%LastMessageDate%>	The date/time the last message in the thread was created.
<%LastPoster%>	The nickname of the user that last posted to this thread.
<%Link%>	The link to the full text.
<%LoggedInUserLevel%>	The level of the user currently accessing the system.
<%LoggedInUserVerified%>	This field is set to 1 if the user currently accessing the system has been verified.
<%MessageID%>	A value representing the current message as a number.
<%NewMessageLink%>	The link to the page for posting a new top level message.
<%PostedBy%>	The numeric user ID.
<%PostedByHost%>	The IP address of the user who posed the message.
<%Published%>	For moderated forums, this is the date/time the moderator published the message. For non-moderated forums this is the date/time the message was created.
<%ReplyMessageLink%>	The link to the page for posting a reply.
<%ShowAllLink%>	A link to a list of messages for this message's group that is not restricted by **MaxDays**.
<%ShowRepliesLink%>	The same as <%Link%>
<%Signature%>	The signature text of the message author.
<%Subject%>	The group name for this message.
<%Theme%>	Theme name for this message.
<%ThemeDescription%>	The theme description for this message.
<%ThemeLink%>	Link to the list of groups.
<%ThreadCount%>	The number of replies to this message.
<%\Title%>	The title of the reply.
<%Today%>	This has a value of 1 when there is a reply to the current message that has been created today.
<%ToggleLockLink%>	A link that will cause the message's locked status to toggle. A locked message will become unlocked and an unlocked message will become locked.

<%TreeLines%>	Lists tree-line characters appropriate to the level of indent for this thread item.
<%UserID%>	An integer ID that represents the author in the system.
<%UserLevel%>	The user level of the author that created the message.
<%UserPostCount%>	The total number of posts the author has made in the system.
<%UserRegDate%>	The date the user first registered in the system.

ForumNavBar.pbo

This object file is used to list a navigation bar, which can help the user quickly navigate through many pages of message data.

<%FirstLink%>	The link to the first page, if it is not currently being viewed.
<%LastLink%>	The link to the last page, if it is not currently being viewed.
<%NavbarItems%>	This field lists all entries.
<%NextLink%>	The link to the next page, if there is a next page.
<%NumPages%>	The number of pages that are required to display all the messages. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PageNum%>	When messages are displayed across several pages, this field contains a numeric value for the page index currently displayed. The option is used when you use **Threadcount** to limit the number of threads on one page.
<%PrevLink%>	The link to the previous page, if there is a previous page.

ForumNavBarItem.pbo

This object file is used to list a navigation bar, which can help the user quickly navigate through many pages of message data.

<%CurrentPage%>	The page number that is displayed for the list of pages.
<%IsCurrent%>	This has a value of 1 if this item represents the current page in the list.
<%PageLink%>	A link to the message represented by this item.
<%\PageNumber%>	The page number of this item.

pbs search

<pbs:search>

You can search all messages in the forums.

The example code is set up so that you search within the scope you are viewing. For example, if you are viewing a group, you will only search within that group.

Template

This function can be placed on all templates.

Options

Encoding	Because the search URL is built using a PBS tag, the output cannot be encoded to ensure compliance with W3C standards. The <pbs:search> tag now accepts a parameter
	
	`encode="1"`
	
	that will encode the returned URL in the same way as using
	
	`$h`
	
	on a variable. For example:
	
	`<pbs:search encode="1">`
	
	will turn an ampersand character "&" into "&"

Search form

The form action must be {{<pbs:search type=forum>}}like this:

```
<form action="<pbs:search type=forum>" method="post">
```

Fields and buttons

Crit	Used to input the search text string.
Group	(Hidden field) Used to limit the search to the group being viewed or to a specified group. Leaving the field empty will make the form search all groups. Example: `<input type="hidden" name="group" value="<pbs:queryparam param=groupid>">` The tag pbs queryparam is used to output the group ID for the group.
Submit	Used to add a submit button.
Theme	(Hidden field) Used to limit the search to the theme being viewed or to a specified theme. Leaving the field empty will make the form search all themes. Example: `<input type="hidden" name="theme" value="<pbs:queryparam param=themeid>">` The tag pbs queryparam is used to output the theme ID for the theme.

URL parameters

SearchType	Used to qualify the search type. For example:
	
	`<SELECT NAME="SearchType" SIZE="1"`
	
	`onchange="SearchForm.action='<pbs:search type=forum>&SearchType=' + this.selectedIndex">`
	`<OPTION VALUE="*" >All Fields`
	`<OPTION VALUE="Body">Body`
	`<OPTION VALUE="Author">Author`
	`<OPTION VALUE="Title">Title`
	`</SELECT>`
	
	It can have one of the following integer values:
	• 1 = Search all fields
	• 2= Search only the body field
	• 3= Search only the author field
	• 4= Search only the title field

pbs searchresult and forums

<pbs:searchresult>

Template

This function uses the template **gen_forumsearch.pbs**.

Options

Count	Sets the number of messages that will be shown on the search result page.
Showref	Enter `showref=1` to enable the output of the body for the message.
Sites	Used to search multiple site codes. Usage: `sites=AA,BB,CC`

SearchResultForum.pbo

This file works in the same way as SearchResult.pbo

SearchResultForumItem.pbo

<%Body%>	The entire message body.
<%Counter%>	Countains the counter value.
<%Description%>	The description of the theme.

<%GroupLink%>	The link to the overview for the group.
<%GroupName%>	The name of the group to which the message belongs.
<%LeadIn%>	The first 250 characters of the message.
<%MessageLink%>	The link to the message.
<%PublishedDate%>	The published date for the message.
<%Subject%>	The description of the group.
<%ThemeLink%>	The link to the overview for the theme.
<%ThemeName%>	The name of the theme to which the message belongs.
<%\Title%>	This field contains the title of the message.

pbs userverify

<pbs:userverify>

When you use user registration, an e-mail must be sent to the user so she can verify that her e-mail address is correct. The system will automatically redirect the user to this page before she can post any messages.

> Before the user verified function will work currently this ini setting needs to be set:
>
> [UserVerify]
> AutomaticLogin=1
>
> DisableLogin=1

Template

Gen_verify.pbs

UserVerificationMail.pbo

<%Accept%>	The link to the acceptance page.
<%ClientHost%>	The host name for the machine that triggered the verification process.
<%Decline%>	A link to decline the verification.
<%Name%>	The name of the person being verified.

Example

```
<P>Hi <%Name%>.
<P>Please click on the link immediately below this paragraph in order to
activate your new forum account on DailyNews.com:
<%Accept%>
<P>If you do not want to activate your account, or if you never requested the account,
please click on the following link:
<%Decline%>
<P>The request to create this account was made from <%ClientHost%>. If you
suspect that someone may have abused your account, please contact
<A HREF="mailto:john.smith@dailynews.com">john.smith@dailynews.com</A>.
```

In addition to using the standard "accept" and "decline" links, an additional parameter can be added (6.6.0sr19), that changes the resulting template file loaded upon selecting either link. By adding "&UseTemplate=1" to the "accept" or "decline" links, you can direct a user to a template called "gen_verifyaccept.pbs" or "gen_verifydecline.pbs" respectively. This allows the user to be taken to a page that reflects either the "accept" or "decline" action clicked from the verification email.

> In version 6.8, &UseTemplate=1 was case sensitive and would not load the gen_verifyaccept.pbs based if the case did not match.

UserVerify.pbo

<%EMailAddress%>	The e-mail address that is being verified.
<%Error%>	An error message if anything went wrong.
<%NextLink%>	A link back to the page that triggered the verification.
<%SendMailError%>	An error message if anything when wrong during the sending of the email.
<%Status%>	The status for the verification. The field is used to output different texts. Please look at the example to see what the different statuses mean.

Example

```
[%{Status=3}
A verification e-mail has been sent to <%EMailAddress%>. In order to use our forums,
we need to verify your email address. When you receive this e-mail, click on the link
provided to verify this account. Once you have verified your account, you can post in
the forums.<br>
You should receive the verification e-mail within the next 5-10 minutes.<br>
%]
[%{Status=1}
Your user account has been verified. You can now proceed to post in our forums. If you
want to modify your user preferences, you can click
<a href="<%pbcs.dll%>/forumoptions?Category=forums">here</a>.<br>
[%If you want to return to the location that triggered the verification process, click
<a href="<%NextLink%>">here</a>.%]<br>
%]
[%{Status=0}
A verification email has already been sent. If you wish to resend the email, click <a
href="userverify?Resend=1">here</a>.
%]
[%{Status=-1}
A problem has occurred. The system cannot find the account you are trying to access.
If you clicked on the link you received through the verification email, please make
sure that your e-mail client did not break up the link. If so, make sure you correct
this problem and retry the operation.<br>
If the problem persists, please contact <a
href="mailto:john.smith@dailynews.com">john.smith@dailynews.com</a>.
%]
[%{Status<-1}
[%<%Error%><br>%]
[%A problem occurred while sending the e-mail. The message was:
<%SendMailError%><br>%]
The status code for this problem is: <b><%Status%></b><br>
%]
[%{Status=2}
Email declined
%]
```