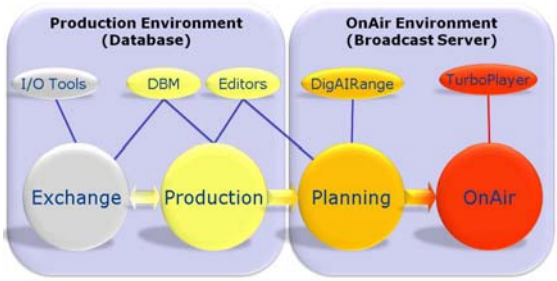
Workflows

OnAir Automation

BroadcastServer

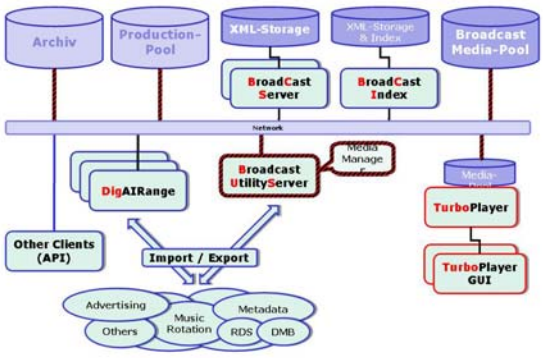
BroadcastServer (BCS) is DigaSystem's highly flexible and central broadcasting service offering powerful interfaces as well as secure redundancy concepts. For security reasons production and playout environment are separated, thus preventing files being changed or deleted by operations during playout.



Like all DigaSystem applications the Broadcast Server System features a modular and scalable architecture. The main part of the system is the BroadcastServer itself. It is based on a XML tree structure offering an enormous variety of features for scheduling, planning and reporting. To make the system highly accessible and operationally secure an optional backup server, called Buddy Server, is available. In case of failure of the main server the system switches over to the Buddy. For that reason a dynamic update between BCS and Buddy happens continuously during normal operation. Clients do not need to restart or modify their settings in case of an emergency.

The third server in a typical Broadcast Server environment is the Broadcast Utility Service (BUS). This server is used for managing audio files, automatically importing rundowns from third party systems as well as for exporting the broadcasted items to web applications or to 3rd party traffic & billing systems for reconciliation.

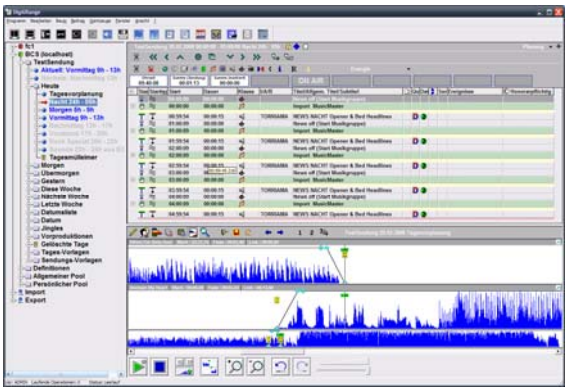
The media management is part of the Broadcast Server's security concept. The BUS copies all scheduled audio files to a central On-Air storage outsides the normal production database and optionally to the local hard disks of the OnAir PCs. In a worst case scenario the On-Air PCs will still be able to broadcast even if they lose their network connection.



Dedicated interfaces integrate 3rd party products like music rotation, advertising disposition or commercial traffic and billing systems. All modules of the Broadcast Server generate detailed operational log files.

Scheduling

The scheduling client DigAIRrange is executed directly from the Database Manager by opening it in the icon bar. DigAIRrange manages the schedules for TurboPlayer.



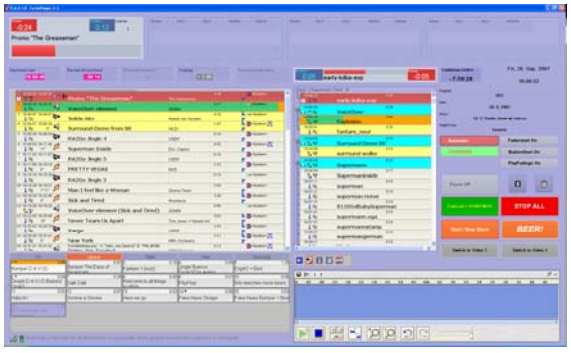
All schedules are based on XML and stored in the BroadcastServer. The content of the schedules can be audio, text, combined audio & text items and also graphics or videos for rich media playout. The screen layout is customizable and can be saved for every production seat. DigAIRrange is able to connect to all BroadcastServers available in the network. DigAIRrange also includes import and export functionality.

TurboPlayer

The real-time OnAir client TurboPlayer is responsible for playout of the rundowns created in DigAIRrange. The screen layout is fully customizable. Every functional area of the screen can be moved, re-sized and coloured regarding the needs of the project. The number of master channels is also configurable. By default, three master channels are used, which allows fully automated Playout including voice tracking. In addition, any number of concurrent jingle channels can be configured. TurboPlayer works with motorized faders or standard analogue mixer consoles.

TurboPlayer allows editing every part of a show. Changes can even be applied to the currently running item, e.g. change the MarkOut to a new fade-out position. All changes are displayed immediately at all workstations including all connected OnAir PCs. The scheduling clients also display which show is currently OnAir, which job is running and which item was skipped or played. TurboPlayer is equipped with a scalable text window with scrolling option which also allows previewing or changing the text.

In order to use more than one TurboPlayer per show a special remote display feature has been implemented showing the items running in another TurboPlayer. This information is displayed in a special area of the screen and keeps all show relevant staff members up to date e.g. by displaying the remaining time of the running item in real time.



TurboPlayer also stores additional information like the real duration of broadcasted items in case it is stopped before the planned end. All these updates are made in real time.