







 Start Event	<pre>context artifact::Initialize() post: Artifact.allInstances()->exists(af af.ocIsNew() and af.end=false and result=af)</pre> Initialize creates a new artifact with its end attribute set to false.
 Artist Selected	<pre>context artifact::ArtistSelected(artist:Artist) post: self.artist=artist</pre> ArtistSelected assigns the artist given as input to the process's artifact.
 Calc Send Tracks	<pre>context artifact::CalcAndSendTracks(): Set(Track) post: result = Track.allInstances()->select(t t.album.artist = self.artist)</pre> CalcSendTracks initially obtains all the tracks (Track.allInstances()) and selects those belonging to an album whose artist is equal to the artist linked to the artifact (t.album.artist = af.artist). It returns this list as a result (hence, the result keyword).
 Tracks PName Selected	<pre>context artifact::TracksPNameSelected(trackL:Set(Track), plName:String) post: self.track=trackL and self.pname=plName</pre> TracksPNameSelected assigns the set of tracks provided as input to the artifact, and stores the playlist name given as input in the corresponding attribute of the artifact.
 Build Playlist	<pre>context artifact::BuildPlaylist() post: Playlist.allInstances()->exists(pl pl.ocIsNew() and pl.name=self.pname and pl.track->includesAll(self.track))</pre> BuildPlaylist creates a new instance of Playlist (ocIsNew). This playlist will have the name stored in the artifact (pl.name = af.pname) and the tracks of the playlist will correspond to the tracks linked to the artifact (pl.track->includesAll(af.track)).
 Send Exist Playlists	<pre>context artifact::SendExistPlaylists(): Set(Playlist) post: result = self.artist.album.track.playlist->asSet()</pre> SendExistPlaylists obtains the playlists that already exist which contain tracks by the selected artist selected and sends this information to the user.
 Playlist Sent	<pre>context artifact::PlaylistSent(): Playlist post: result = self.playlist</pre> PlaylistSent returns the playlist that has been created (the one assigned to the artifact) as a result.
 Continue	<pre>context artifact::Continue(e:bool) post: self.end=e</pre> Continue updates the value of attribute end in the artifact with the given input.
 End Event	<pre>context artifact::End() post: Artifact.allInstances()->excludes(self)</pre> End deletes the artifact linked to this instance of the process and all the relationships it takes part in.