War God Lore Pack

Technical Reference

Version 1.0.0

For Unreal Engine 5.3+ | Generated February 04, 2026

Overview

This technical reference provides detailed information about the War God Lore Pack's structure, data formats, and integration patterns for developers working with Unreal Engine.

Pack Statistics

Asset Type	Count	Format
Voice Lines (Dialogue)	566	Audio + DataTable
Voice Effects	42	Audio + DataTable
Songs	40	Audio + DataTable
Character Profile Entries	1	DataTable (CSV)
Written Content	85	DataTable (CSV)

Data Structures

Dialogue DataTable Schema

The DT_Dialogue DataTable uses the following structure:

Column	Type	Description
Name	FName (Row Key)	Unique identifier for the dialogue entry
DialogueName	FString	Human-readable name for the dialogue
ResponseText	FString	The actual dialogue text spoken
CharacterName	FString	Name of the speaking character
EmotionalTone	FString	Primary emotional category (Combat, Social, etc.)
ContextTags	FString	Hierarchical tags: category/subcategory/size
NPCType	FString	Character archetype identifier
VoiceAudio	TSoftObjectPtr<USoundWave>	Reference to the audio asset

ContextTags Format

Context tags follow the pattern: `category/subcategory/size`

```
Examples:
combat/battle_cry/sm       - Short battle cry
social/greeting/md         - Medium-length greeting
story/dragon/xl            - Extended dragon story
emotion/anger/lg           - Long angry expression
```

Character Profile Schema

Column	Type	Description
Name	FName	Row identifier
CharacterName	FString	Display name of the character
Backstory	FString	Full character backstory narrative
Personality	FString	Personality traits and speech patterns
Motivations	FString	Character goals and driving forces

Written Content Schema

Column	Type	Description
Name	FName	Row identifier
ContentTitle	FString	Type of written content
BodyText	FString	The full text content
ContentType	FString	Format type (written)
ContextTags	FString	Category tags: category/subcategory
WordCount	int32	Approximate word count

Written Content Types

- **note_reminder** - Quick reminder notes
- **note_warning** - Warning messages left for others
- **note_recipe** - Potion and crafting recipes
- **note_research** - Research observations
- **letter_authority** - Formal correspondence
- **letter_friend** - Casual letters to friends
- **journal_recent** - Recent journal entries

- **journal_old** - Historical journal entries
- **book_philosophy** - Philosophical treatises
- **book_history** - Historical accounts
- **poetry_nature** - Nature-themed poetry
- **poetry_war** - War-themed poetry

Folder Structure

```
Deity_Male_Pack/
├── Audio/
│   └── Dialogue/
│       ├── Combat/
│       │   ├── Battle_Cry/
│       │   ├── Enemy_Spotted/
│       │   └── ...
│       ├── Social/
│       │   ├── Greeting/
│       │   ├── Farewell/
│       │   └── ...
│       ├── Emotion/
│       ├── Story/
│       └── ...
└── DataTables/
    ├── DT_Dialogue_deity_male.uasset
    ├── DT_CharacterProfile_deity_male.uasset
    ├── DT_Quests_deity_male.uasset
    └── DT_WrittenContent_deity_male.uasset
```

Dialogue Categories

Voice lines are organized into categories for easy integration. Use the ContextTags field to filter dialogue by situation.

Category	Description
Combat	Battle cries, enemy callouts, tactical commands
Taunt	Intimidation, challenges, victory declarations
Death	Dying words, resurrection reactions
Physical	Jumping, climbing, exertion sounds
Pain	Damage reactions from hits
Emotion	Joy, anger, sadness, fear expressions
Response	Yes, no, understood, quick responses
Social	Greetings, farewells, social interactions
Self	Idle muttering, internal thoughts
Commentary	Location-based observations
Weather	Reactions to weather conditions
Discovery	Reactions to finding items, secrets
Story	Narrative tales and stories

Integration Patterns

Blueprint Integration

Example: Playing a random greeting when player approaches NPC:

```
// In NPC Blueprint - On Player Overlap Event

1. Get DataTable Reference: DT_Dialogue_deity_male
2. Get All Row Names from DataTable
3. Filter rows where ContextTags contains "social/greeting"
4. Get Random Index from filtered array
5. Get Row by Name from DataTable
6. Play Sound 2D or 3D using VoiceAudio reference
7. Optionally display ResponseText in UI
```

C++ Integration

```
// Load and query dialogue
UDataTable* DialogueTable = LoadObject<UDataTable>(nullptr,
    TEXT("/Game/Deity_Male_Pack/DataTables/DT_Dialogue_deity_male"));

TArray<FDialogueRow*> AllRows;
DialogueTable->GetAllRows<FDialogueRow>(TEXT(""), AllRows);

// Filter by context
TArray<FDialogueRow*> CombatLines;
for (FDialogueRow* Row : AllRows)
{
    if (Row->ContextTags.Contains(TEXT("combat")))
    {
        CombatLines.Add(Row);
    }
}

// Play random line
int32 Index = FMath::RandRange(0, CombatLines.Num() - 1);
PlaySound(CombatLines[Index]->VoiceAudio);
```

Performance Considerations

Audio Memory

- Audio files are provided in high-quality format
- Consider converting to compressed format for mobile targets
- Use audio streaming for longer lines (XL category)
- Pre-cache commonly used categories at level load

DataTable Access

- Cache DataTable references rather than loading repeatedly
- Pre-filter rows by category at initialization
- Use row name lookups for known dialogue IDs

Version History

Version	Date	Changes
1.0.0	February 2026	Initial release