

ContextResolver

September 4, 2026

ContextResolver

Kind: Class

Source: [atloria-monorepo/packages/doc-automation/src/v2/stages/stage-10-generation/context-resolver.ts](https://github.com/atloria-monorepo/packages/doc-automation/src/v2/stages/stage-10-generation/context-resolver.ts)

ContextResolver - Resolves cross-chunk entity references

When generating documentation for a document, we may need entities that exist in OTHER chunks. This resolver fetches them from the global entity map without loading everything into memory.

Methods

Method	Signature	Returns
<code>setEntityMap</code>	<code>setEntityMap(entityMap: Map<string, Entity>)</code>	<code>void</code>
<code>resolveForPlan</code>	<code>resolveForPlan(plan: DocumentPlan)</code>	<code>Promise<Entity[]></code>
<code>resolveByName</code>	<code>resolveByName(entityName: string)</code>	<code>`Entity</code>
<code>resolveById</code>	<code>resolveById(entityId: string)</code>	<code>`Entity</code>
<code>handleDeepLoadRequest</code>	<code>handleDeepLoadRequest(request: DeepLoadRequest)</code>	<code>Promise<DeepLoadResponse></code>
<code>getByFilePath</code>	<code>getByFilePath(pattern: string)</code>	<code>Entity[]</code>
<code>getByType</code>	<code>getByType(type: Entity['type'])</code>	<code>Entity[]</code>
<code>clearCache</code>	<code>clearCache()</code>	<code>void</code>
<code>getCacheStats</code>	<code>getCacheStats()</code>	<code>{ size: number; hits: string[] }</code>

Method	Signature	Returns
hasEntity	hasEntity(entityId: string)	boolean
getEntityCount	getEntityCount()	number

Where it refuses work

- ContextResolver stops the work with an early return when `cached?.found` .
- ContextResolver stops the work with an early return when `entity.name === entityName` .
- ContextResolver stops the work with an early return when `this.deepLoadedCache.has(cacheKey)` .