

```
# Req6-1~6-3 vs Req6-4 Proposal – Comparison
```

```
> Generated: 2026-02-02 (UTC)
```

Summary

- **Req6-1~6-3 (implemented)**: provide *downward query* (structural + expanded) and *upward paths* (segments), with deep bbox and TooManyResults guardrails.
- **Req6-4 (proposal)**: add *detailed path edges* and *direct parent/child queries* to make debugging and fanin/fanout analysis practical.

Comparison Table

Item	Req6-1 (query_down structural)	Req6-2 (query_up_paths)	Req6-3 (query_down expanded)	Req6-4 (proposal)
Primary use	From a root cell, list instance records by depth (arrays not expanded)	From TOP, list **paths** to a target cell (segments)	From a root cell, list instance records by depth (**arrays expanded**) Debug/analysis: detailed paths + direct parent/child queries	
Direction	Down	Up (from TOP)	Down	Up + local (parents/children)
Array handling	1 record per array instance	Path traversal sees instance edges; does not expose per-array element	Expands regular arrays to per-element records with `expanded_index`	Proposal includes both: detailed edges and (optionally) expanded variants
Output core	`instances[]` records	`paths[]` (segments)	`instances[]` records (expanded)	`paths[]` with **edge details** + `parents[]` / `children[]` convenience endpoints
Path context	Each record has `path` (segments) from query root to the **parent cell** containing the instance	Each path is `segments=[gds, TOP, ..., target]`	Each record has `path` (segments) from query root to the **parent cell**	Proposal adds **edge list** per path (each edge carries instance metadata)
BBox	Deep bbox (all layers) per instance record	Not included	Deep bbox (all layers) per expanded element record	Required; especially useful on per-edge records and parent/child queries
TooManyResults guardrail	`limit` (default 10k) → error `TooManyResults` with clear message	`max_paths` (default 10k) → error `TooManyResults` `limit` (default 10k) → error `TooManyResults` Applies to `max_paths` / `limit` similarly; requires clear message/comment		
Multi top cell	Not required (root is explicit cell)	**Error** if multiple/no top cell	Not required (root is explicit cell)	For TOP-based queries: still **error**; for local queries (parents/children) can be independent (needs final decision)
Typical questions answered	"From this cell, what do I instantiate within N levels?" "Where is this cell used (as a path from TOP)?" "How many physical instance elements exist under this root (bounded by limit)?" "Show me the exact chain and transforms/bboxes along each path" + "Who are my direct parents/children?"			

Req6-4 Proposed Methods (draft)

```
### 6-4.1 `hier.query_up_paths_detailed`  
- Input: `cell` (target), `mode` (structural/expanded), `max_paths`  
- Output: list of paths, each containing:  
  - `segments`: `[gds, TOP, ..., target]`  
  - `edges`: per-step instance descriptors (parent-child), including trans/array/bbox (+ expanded_index when applicable)
```

```
### 6-4.2 `hier.query_parents`  
- Input: `cell`, `mode`, `limit`
```

- Output: direct parents with instance metadata.
- ### 6-4.3 `hier.query_children`
- Input: `cell`, `mode`, `limit`
 - Output: direct children with instance metadata.