```
from swarms.structs.agent import Agent
from typing import List



def showcase_available_agents(
    agents: List[Agent],
    name: str = None,
    description: str = None,
    format: str = "XML",
) -> str:
    """
    Format the available agents in either XML or Table format.

    Args:
        agents (List[Agent]): A list of agents to represent
        name (str, optional): Name of the swarm
        description (str, optional): Description of the swarm
        format (str, optional): Output format ("XML" or "Table"). Defaults to "XML"

    Returns:
        str: Formatted string containing agent information
    """

    def truncate(text: str, max_length: int = 130) -> str:
        return (
            f"{text[:max_length]}..."
```

```
        if len(text) > max_length
        else text
    )


output = []


if format.upper() == "TABLE":
    output.append("\n| ID | Agent Name | Description |")
    output.append("|-----|------------|-------------|")
    for idx, agent in enumerate(agents):
        if isinstance(agent, Agent):
            agent_name = getattr(agent, "agent_name", str(agent))
            description = getattr(
                agent,
                "description",
                getattr(
                    agent, "system_prompt", "Unknown description"
                ),
            )
            desc = truncate(description, 50)
            output.append(
                f"| {idx + 1} | {agent_name} | {desc} |"
            )
        else:
            output.append(
                f"| {idx + 1} | {agent} | Unknown description |"
```

```
            )
    return "\n".join(output)


# Default XML format
output.append("<agents>")
if name:
    output.append(f"  <name>{name}</name>")
if description:
    output.append(
        f"  <description>{truncate(description)}</description>"
    )
for idx, agent in enumerate(agents):
    output.append(f"  <agent id='{idx + 1}'>")
    if isinstance(agent, Agent):
        agent_name = getattr(agent, "agent_name", str(agent))
        description = getattr(
            agent,
            "description",
            getattr(
                agent, "system_prompt", "Unknown description"
            ),
        )
        output.append(f"    <name>{agent_name}</name>")
        output.append(
            f"    <description>{truncate(description)}</description>"
        )
```

```
    else:
        output.append(f"    <name>{agent}</name>")
        output.append(
            "    <description>Unknown description</description>"
        )
    output.append("  </agent>")
output.append("</agents>")

return "\n".join(output)
```